

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE INFORMÁTICA



UNIVERSIDAD CARLOS III DE MADRID

Proyecto Fin de Carrera

Ingeniería Técnica en Informática de Gestión

" SISTEMA INTERMEDIARIO DE GESTION DE PRESTAMOS DE  
OBRAS DE ARTE ENTRE MUSEOS "

**AUTOR:** Matei García Udrea.

**TUTOR:** Jose María Sierra Cámara



## ÍNDICE DE CONTENIDOS

### Contenido

1 - INTRODUCCIÓN .....	5
<b>1.1 - INTRODUCCIÓN AL PROYECTO</b> .....	5
<b>1.3- MOTIVACIÓN</b> .....	7
<b>1.4 - ENFOQUE DEL PROYECTO</b> .....	8
<b>1.5 - OBJETIVOS</b> .....	9
<b>1.6 - ESTRUCTURA DE LA MEMORIA</b> .....	12
2 - TECNOLOGÍAS USADAS .....	13
<b>2.1 - DESCRIPCIÓN DE GLASSFISH</b> .....	13
<b>2.2 – JAVA</b> .....	14
2.2.1 Orientado a Objetos .....	15
2.2.2 Independencia de la plataforma .....	16
<b>2.3 - ORACLE</b> .....	18
2.3.1 Gestión de transacciones .....	18
2.3.2 Escalabilidad .....	19
<b>2.4 - POSTFIX</b> .....	20
<b>2.5 – ETIQUETAS PERSONALIZADAS (CUSTOMIZED TAGS)</b> .....	21
<b>2.6 – PRINCIPALES FICHEROS DEL SISTEMA</b> .....	22
3 - ANÁLISIS DE REQUISITOS Y DISEÑO DEL PORTAL .....	23
<b>3.1 – DEFINICIÓN DE REQUISITOS</b> .....	24
<b>3.2 - IDENTIFICACIÓN DE REQUISITOS</b> .....	25
<b>3.3 - REQUISITOS DE USUARIO</b> .....	28
3.3.1 – Descripción General .....	28
3.3.2 – Requisitos de Usuario de Capacidad .....	30
3.3.3 – Requisitos de Usuario de Restricción .....	35
4 – DIAGRAMA DE CASOS DE USO GENERAL .....	38
<b>4.1 – Diagrama de casos de uso para el usuario anónimo</b> .....	39
<b>4.2 – Diagrama de casos de uso para el usuario registrado</b> .....	40
<b>4.3 – Diagrama de casos de uso para el validador</b> .....	42



5 - DIAGRAMA DE ARQUITECTURA.....	45
A continuación vamos a indicar la arquitectura general del sistema, y además vamos a	
indicar mediante un ejemplo como funcionaría dicha arquitectura cuando se realiza una	
consulta sobre el Sistema Intermediario.....	45
<b>5.1 Modelo de Datos .....</b>	<b>46</b>
<b>5.2 Funcionamiento general del Sistema .....</b>	<b>47</b>
<b>5.3 Arquitectura del sistema .....</b>	<b>50</b>
5.3.1 Modelo Vista Controlador.....	51
5.3.2 Utilización del Modelo Vista Controlador en el proyecto.....	52
5.3.3 Patrón Bean .....	53
5.3.4 Utilización del Patrón Bean en el proyecto.....	53
6 - DIAGRAMAS DE SECUENCIA .....	54
6.01 – Diagrama de Secuencia “DS001” Para el Caso de Uso “CU001” .....	55
6.02 - Diagrama de Secuencia “DS002” Para el Caso de Uso “CU002” .....	56
6.03a - Diagrama de Secuencia “DS003a” Para el Caso de Uso “CU003” .....	57
6.03b - Diagrama de Secuencia “DS003b” Para el Caso de Uso “CU003” .....	58
6.04 - Diagrama de Secuencia “DS004” Para el Caso de Uso “CU004” .....	59
6.05 - Diagrama de Secuencia “DS005” Para el Caso de Uso “CU005” .....	61
6.06 - Diagrama de Secuencia “DS006” Para el Caso de Uso “CU006” .....	62
6.07a - Diagrama de Secuencia “DS007a” Para el Caso de Uso “CU007” .....	63
6.07b - Diagrama de Secuencia “DS007b” Para el Caso de Uso “CU007” .....	65
6.08 - Diagrama de Secuencia “DS008” Para el Caso de Uso “CU008” .....	67
6.09 - Diagrama de Secuencia “DS009” Para el Caso de Uso “CU009” .....	69
6.10 - Diagrama de Secuencia “DS010” Para el Caso de Uso “CU010” .....	71
6.11a - Diagrama de Secuencia “DS011a” Para el Caso de Uso “CU011” .....	72
6.11b - Diagrama de Secuencia “DS011b” Para el Caso de Uso “CU011” .....	73
6.12a - Diagrama de Secuencia “DS0012a” Para el Caso de Uso “CU0012” .....	74
6.12b - Diagrama de Secuencia “DS0012b” Para el Caso de Uso “CU0012” .....	75
7 - TRAZABILIDAD.....	76
<b>7.1 – Tabla de Trazabilidad .....</b>	<b>77</b>
8 - DISEÑO GRÁFICO DE LA APLICACIÓN WEB .....	78
<b>8.01 - DISEÑO DE LA PÁGINA PRINCIPAL.....</b>	<b>78</b>
<b>8.02 - DISEÑO DE LA PÁGINA PRINCIPAL DEL USUARIO REGISTRADO .....</b>	<b>79</b>
<b>8.03 - DISEÑO DE LA PÁGINA PRINCIPAL DEL VALIDADOR .....</b>	<b>80</b>
<b>8.04 - DISEÑO DEL MENÚ PRINCIPAL.....</b>	<b>81</b>
<b>8.05 - DISEÑO DEL FORMULARIO DE ACCESO.....</b>	<b>83</b>



<b>8.06 - DISEÑO DEL MENÚ DE BIENVENIDA.....</b>	<b>83</b>
<b>8.07 - DISEÑO DE MENÚ INFERIOR.....</b>	<b>85</b>
<b>9 - BATERÍA DE PRUEBAS REALIZADAS AL SISTEMA.....</b>	<b>86</b>
<b>9.1 Pruebas de Usuario.....</b>	<b>86</b>
<b>9.2 Pruebas de Validador.....</b>	<b>95</b>
<b>10 - CONCLUSIONES Y TRABAJOS FUTUROS.....</b>	<b>101</b>
<b>10.1 Conclusiones.....</b>	<b>101</b>
<b>10.2 Trabajos Futuros.....</b>	<b>101</b>
<b>ANEXO 1 – PRESUPUESTO Y PLANIFICACIÓN .....</b>	<b>103</b>
<b>Anexo 1.1 Gantt del Proyecto Diagramas .....</b>	<b>103</b>
<b>Anexo 1.2 Gantt del Proyecto Tareas.....</b>	<b>109</b>
<b>Anexo 1.3 Presupuesto del Proyecto.....</b>	<b>110</b>
<b>ANEXO 2 – MANUAL DE USUARIO .....</b>	<b>111</b>
<b>Usuario Registrado.....</b>	<b>111</b>
<b>Validador .....</b>	<b>116</b>



# 1 - INTRODUCCIÓN

## 1.1 - INTRODUCCIÓN AL PROYECTO

Este proyecto es un sistema realizado para la correcta organización y control de los diferentes préstamos de obras de arte que se realizan entre los museos. En concreto, se trata de crear un único sistema, por el cual un museo pueda consultar y solicitar préstamos de una o varias obras de arte a cualquier museo que esté dado de alta en el mismo.

Para ello se ha optado por una aplicación web realizada mediante java (J2EE) para facilitar su puesta en producción en cualquier servidor, valiéndonos de la principal característica de java que es su portabilidad, puesto que es independiente de la arquitectura.

Asimismo para las BBDD de los museos, se ha optado por Oracle 10g Express edition, puesto que ofrece a la vez la robustez de Oracle, y es totalmente gratuito.

Con respecto al diseño y a la implementación se ha optado por utilizar un patrón modelo vista-controlador, para poder implementar fácilmente cualquier desarrollo posterior que pueda ser requerido por el cliente, teniendo además en cuenta la seguridad y flexibilidad que ofrece el mismo.

Es importante adjuntar una documentación completa y de fácil entendimiento sobre el manejo, a todos los niveles, de dicha aplicación. Por ello, en los apéndices se adjunta la documentación tanto de manejo a nivel de usuario como de validador del portal web.



## 1.2 – DESCRIPCIÓN PRELIMINAR DEL PROBLEMA

Actualmente, para la petición de préstamos de obras entre diferentes museos se contacta bien vía internet o bien mediante teléfono o correo electrónico en qué museo está una obra.

A continuación contactan vía teléfono o correo electrónico con un responsable y acuerdan el préstamo para una fecha determinada y luego se intercambian los papeles oficiales

Este procedimiento se repite con tantas obras como conste una exposición temporal. En algunos casos el número de obras puede ascender a 300-400 obras por exposición temporal, lo que supone un gran trabajo y coste.

Además, los canales de información, al no ser dedicados exclusivamente a este fin no son plenamente seguros (los teléfonos pueden ser pinchados y los mails interceptados).

Por si fuera poco, debido a la propia naturaleza de la comunicación humana, es previsible que se puedan producir fallos en lo que a entrega y recogida de obras se refiere (fallos de dirección, fallos de entrega... Dichos fallos pueden llegar a retrasar o incluso arruinar una exposición.

Por todo ello, se va a realizar una aplicación web para la gestión de los préstamos de obras de arte entre diferentes museos. Sus principales características serán:

- Una aplicación dedicada: Su único fin será propiciar la comunicación entre los diferentes museos para el préstamo de obras de arte
- Segura: La aplicación estará protegida en un servidor y para poder acceder al sistema se hará mediante login/password, pudiéndose utilizar por supuesto otras medidas de seguridad como los certificados digitales.
- Rápida: Para lo que antiguamente se necesitaban varias horas o incluso días, ahora es cuestión de minutos. Mediante unos pocos clicks, se solicitan y se pueden llegar a validar cientos de peticiones
- De fácil manejo: La aplicación es muy intuitiva y sencilla de utilizar. Se incluye además en la memoria un manual de usuario donde se explica cómo utilizar la misma
- Robusta: La aplicación estará realizada en JAVA 1.6, será independiente de la plataforma donde se quiera instalar y será robusta y resistente a cualquier posible incidencia.



### **1.3- MOTIVACIÓN**

La motivación de este proyecto, es implementar y definir un sistema capaz de cumplir con los requisitos anteriormente descritos por el cliente, teniendo en cuenta sus necesidades de facilidad de acceso y de manejo, pero sin olvidar el aspecto fundamental de la seguridad. Además debido a la arquitectura flexible que se utiliza, es muy sencillo realizar cualquier modificación o evolución del mismo. Por último indicar que el nivel de reutilización del código para aplicaciones similares es muy alto.

Con respecto al diseño e implementación de la aplicación, se ha hecho hincapié en cómo se ha creado la misma en base a los requisitos software planteados y en el diseño se hace una descripción de cómo generarlo con las tecnologías usadas, de modo que sea fácil reproducir los resultados obtenidos en este proyecto para futuros desarrollos.

Debido a que la aplicación está orientada a organismos públicos es importante adjuntar una documentación completa y de fácil entendimiento sobre el manejo, a todos los niveles, de dicha aplicación. Por ello, en los apéndices se adjunta la documentación tanto de manejo a nivel de usuario como de validador de la misma.



## 1.4 - ENFOQUE DEL PROYECTO

En este proyecto se va a realizar la implementación del sistema anteriormente mencionado, para lo cual tenemos que tener en cuenta varios aspectos fundamentales:

- 1) Facilidad de manejo: El sistema no puede ser muy complicado de manejar para un usuario final
- 2) Fiabilidad: El sistema debe ser perfectamente fiable, es muy importante no perder información. En apartados posteriores, comentaremos la necesidad de hacer backups periódicos de las BBDD de los museos
- 3) Seguridad: Bajo ninguna circunstancia se puede permitir el acceso a usuarios no deseados al sistema o a la información del mismo

Por ello, la configuración se ha realizado con el siguiente que permita cumplir con esos requisitos:

- GlassFish (Servidor Web).
- Oracle 10g Express Edition (Sistema Gestor de Base de Datos).
- Java (Lenguaje de programación para la gestión del Sistema Intermediario).
- Servidor de correo Postfix (Para el envío automático de mails)
- Customized tags (Librerías Java para la gestión y manejo de etiquetas personalizadas)

Se ha decidido utilizar Java debido principalmente a su característica de independencia de la plataforma. Además el despliegue en cualquier tipo de servidor es muy sencillo debido al formato War. Se utilizan etiquetas personalizadas en los JSPs de la vista para evitar el uso del código java en los mismos y así potenciar la reutilización de código.





## 1.5 - OBJETIVOS

Como ya se ha dicho, los objetivos son la creación de un sistema intermediario de fácil uso para la gestión, consulta y solicitud de prestamos de obras de arte entre los mismos. Desglosando un poco esos objetivos podríamos dividirlos del siguiente modo:

- **Estudio de requisitos software** planteados para el sistema. Es necesario realizar un estudio de requisitos software para conocer las necesidades reales de los usuarios (museos) que van a utilizar la aplicación. Por medio de estos requerimientos, se podrá hacer un estudio (en conjunción con las propiedades que ofrecen las herramientas a usar), para así determinar cómo debe llevarse a cabo el desarrollo del proyecto.
- **Selección de herramientas se van a usar** para acometer los requisitos especificados. GlassFish, Oracle 10g Express Edition, Java, Postfix, Etiquetas Personalizadas. Se ha optado por estas herramientas (como se explicará más adelante) básicamente por ser o software libre o gratuitas y porque son de una calidad contrastada. Además, con el uso de ellas podrán afrontarse todos los requisitos planteados de la forma más satisfactoria posible.
  - GlassFish (Servidor Web Gratuito avalado por SUN MICROSYSTEMS).
  - Oracle 10g Express Edition (Sistema Gestor de Base de Datos gratuito y de gran potencia).
  - Java (Lenguaje de programación mas utilizado actualmente. Ideal para la gestión del Sistema Intermediario).
  - Servidor de correo Postfix (Servidor de correo Linux, gratuito, muy potente y de fácil manejo)
  - Customized tags(JSTL) (Librerías Java para la gestión y manejo de etiquetas personalizadas)



➤ **Diseño de la aplicación.**

- Estándar de Ingeniería del Software “ESA PSS-05-0”: utilizado por la Agencia Espacial Europea. Dicho estándar puede consultarse en la página web oficial de la Agencia Espacial Europea, en la siguiente URL de Internet: <http://www.esa.int/esaCP/index.html>
- Toma de Requisitos: En la fase de toma de requisitos de usuario, o fase de definición del problema, se definirán el ámbito y el alcance del sistema, es decir, lo que se espera que haga el sistema. Los requisitos de usuario son de dos tipos: de capacidad y de restricción.
- Diagramas de Casos de Uso: Explica gráficamente la secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.
- Diagrama de Arquitectura: Muestra gráficamente la estructura y relación de las tecnologías software utilizadas en el proyecto.
- Diagramas de secuencia: Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase.
- Diseño interfáz gráfico de la aplicación: acorde a los requerimientos. Una vez que se tiene realizado un estudio de lo que se necesita y de quién y cómo lo va a usar, se podrá realizar un diseño de la aplicación. Debido a la necesidad de mantener un único estilo para toda la aplicación, se va a utilizar principalmente una única hoja de estilos css definida para todo el proyecto. Las únicas excepciones serán las páginas principales, donde se añadirán extras para mejorar el diseño de las mismas.



➤ **Implementación de la aplicación.**

- Implementación de la aplicación: La implementación de la misma se hará principalmente haciendo uso de la potencia que proporciona Java. Se utilizarán sus librerías para poder acceder a las BBDD en remoto de cada uno de los museos, y se modificarán los datos correspondientes en los mismos.

➤ **Creación de un Manual de Usuario** para la utilización del mismo por cualquier persona implicada en el sistema

➤ **Realizar una serie de pruebas** para verificar el correcto cumplimiento de todos los requisitos del sistema.

➤ **Exponer una serie de líneas futuras de mejora e implementaciones** para añadir funcionalidades al sistema. Es un objetivo de este proyecto el plantear una serie de ampliaciones y mejoras que podrán implementarse en un futuro. Estas ideas se expondrán con más detalle en su correspondiente sección.

➤ **Hacer un estudio de tiempos y presupuestos sobre el trabajo realizado.** Para finalizar los objetivos, planteamos la realización de un estudio (usando Project) con el que poder analizar las diferentes fases de las que éste ha constando y hacer una representación esquemática de la duración de esas fases. Esto es bastante útil, ya que con esa información uno se puede hacer a la idea de los tiempos que supone hacer una implementación de las características que tiene este proyecto.



## 1.6 - ESTRUCTURA DE LA MEMORIA

La memoria tiene tres partes diferenciadas que describiremos a continuación para así facilitar su lectura y poder entender mejor el porqué de la distribución escogida.

- 1) En la primera, se tratan los aspectos de configuración de los diferentes elementos que componen el sistema, tales como: el servidor web Glassfish, la configuración de la BBDD, etc. Es, por tanto, la parte con la que conseguimos dejar un sistema listo para poder implementar la aplicación. A continuación se encarga de describir configuraciones de ficheros, directivas y hace mención al aspecto de la seguridad.
- 2) En la segunda parte, se tratan los aspectos de diseño e implementación del portal. Se hará, en primer, lugar un diseño por medio de diagramas UML, casos de Uso y Diagramas de Secuencia. Luego, se comentará diseño gráfico de las aplicaciones y, finalmente, se describirá la implementación.
- 3) En la tercera parte, se abordan los temas finales, un estudio del desarrollo del proyecto, con el que podremos ver las diferentes fases y su duración, una batería de pruebas con la que comprobar el correcto funcionamiento de todo lo implementado y configurado, una definición de posibles líneas futuras a implementar para la continuación de este proyecto y, un manual de usuario, especialmente enfocado a la utilización de nuestra aplicación web tanto por parte de usuarios como validadores.



## 2 - TECNOLOGÍAS USADAS

A continuación vamos a describir las tecnologías que utilizamos en el proyecto. Además al final de este punto haremos una guía de los principales ficheros utilizados en el mismo.

### 2.1 - DESCRIPCIÓN DE GLASSFISH

GlassFish es un servidor de aplicaciones desarrollado por Sun Microsystems que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. La versión comercial es denominada Sun GlassFish Enterprise Server. Es gratuito y de código libre, se distribuye bajo un licenciamiento dual a través de la licencia CDDL y la GNU GPL.

GlassFish está basado en el código fuente donado por Sun y Oracle Corporation, éste último proporcionó el módulo de persistencia TopLink. GlassFish tiene como base al servidor Sun Java System Application Server de Sun Microsystems, un derivado de Apache Tomcat, y que usa un componente adicional llamado Grizzly que usa Java NIO para escalabilidad y velocidad.



## 2.2 – JAVA

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrollados por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar los programas Java todavía no es software libre).

Las dos características más importantes de Java son:

- Es un lenguaje Orientado a Objetos
- Es un lenguaje Independiente de la plataforma



### 2.2.1 Orientado a Objetos

La primera característica, orientado a objetos (“OO”), se refiere a un método de programación y al diseño del lenguaje. Aunque hay muchas interpretaciones para OO, una primera idea es diseñar el software de forma que los distintos tipos de datos que usen estén unidos a sus operaciones. Así, los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos. Un objeto puede verse como un paquete que contiene el “comportamiento” (el código) y el “estado” (datos). El principio es separar aquello que cambia de las cosas que permanecen inalterables. Frecuentemente, cambiar una estructura de datos implica un cambio en el código que opera sobre los mismos, o viceversa. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad y reduciendo el número de proyectos fallidos. Otra de las grandes promesas de la programación orientada a objetos es la creación de entidades más genéricas (objetos) que permitan la reutilización del software entre proyectos, una de las premisas fundamentales de la Ingeniería del Software. Un objeto genérico “cliente”, por ejemplo, debería en teoría tener el mismo conjunto de comportamiento en diferentes proyectos, sobre todo cuando estos coinciden en cierta medida, algo que suele suceder en las grandes organizaciones. En este sentido, los objetos podrían verse como piezas reutilizables que pueden emplearse en múltiples proyectos distintos, posibilitando así a la industria del software a construir proyectos de envergadura empleando componentes ya existentes y de comprobada calidad; conduciendo esto finalmente a una reducción drástica del tiempo de desarrollo. Podemos usar como ejemplo de objeto el aluminio. Una vez definidos datos (peso, maleabilidad, etc.), y su “comportamiento” (soldar dos piezas, etc.), el objeto “aluminio” puede ser reutilizado en el campo de la construcción, del automóvil, de la aviación, etc.

La reutilización del software ha experimentado resultados dispares, encontrando dos dificultades principales: el diseño de objetos realmente genéricos es pobremente comprendido, y falta una metodología para la amplia comunicación de oportunidades de reutilización. Algunas comunidades de “código abierto” (open source) quieren ayudar en este problema dando medios a los desarrolladores para diseminar la información sobre el uso y versatilidad de objetos reutilizables y bibliotecas de objetos.



### 2.2.2 Independencia de la plataforma

La segunda característica, la independencia de la plataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Este es el significado de ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo, tal como reza el axioma de Java, “write once, run everywhere”.

Para ello, se compila el código fuente escrito en lenguaje Java, para generar un código conocido como “bytecode” (específicamente Java bytecode)—instrucciones máquina simplificadas específicas de la plataforma Java. Esta pieza está “a medio camino” entre el código fuente y el código máquina que entiende el dispositivo destino. El bytecode es ejecutado entonces en la máquina virtual (JVM), un programa escrito en código nativo de la plataforma destino (que es el que entiende su hardware), que interpreta y ejecuta el código. Además, se suministran bibliotecas adicionales para acceder a las características de cada dispositivo (como los gráficos, ejecución mediante hebras o threads, la interfaz de red) de forma unificada. Se debe tener presente que, aunque hay una etapa explícita de compilación, el bytecode generado es interpretado o convertido a instrucciones máquina del código nativo por el compilador JIT (Just In Time).

Hay implementaciones del compilador de Java que convierten el código fuente directamente en código objeto nativo, como GCJ. Esto elimina la etapa intermedia donde se genera el bytecode, pero la salida de este tipo de compiladores sólo puede ejecutarse en un tipo de arquitectura.

La licencia sobre Java de Sun insiste que todas las implementaciones sean “compatibles”. Esto dio lugar a una disputa legal entre Microsoft y Sun, cuando éste último alegó que la implementación de Microsoft no daba soporte a las interfaces RMI y JNI además de haber añadido características “dependientes” de su plataforma. Sun demandó a Microsoft y ganó por daños y perjuicios (unos 20 millones de dólares) así como una orden judicial forzando la acatación de la licencia de Sun. Como respuesta, Microsoft no ofrece Java con su versión de sistema operativo, y en recientes versiones de Windows, su navegador Internet Explorer no admite la ejecución de applets sin un conector (o plugin) aparte. Sin embargo, Sun y otras fuentes ofrecen versiones gratuitas para distintas versiones de Windows.

Las primeras implementaciones del lenguaje usaban una máquina virtual interpretada para conseguir la portabilidad. Sin embargo, el resultado eran programas que se ejecutaban comparativamente más lentos que aquellos escritos en C o C++. Esto hizo que Java se ganase una reputación de lento en rendimiento. Las implementaciones recientes de la JVM dan lugar a programas que se ejecutan considerablemente más rápido que las versiones antiguas, empleando diversas técnicas, aunque sigue siendo mucho más lento que otros lenguajes.

La primera de estas técnicas es simplemente compilar directamente en código nativo como hacen los compiladores tradicionales, eliminando la etapa del bytecode. Esto da lugar a un gran rendimiento en la ejecución, pero tapa el camino a la portabilidad. Otra técnica, conocida como compilación JIT (Just In Time, o “compilación al vuelo”), convierte el bytecode a código nativo cuando se ejecuta la aplicación. Otras máquinas virtuales más sofisticadas usan una “recompilación dinámica” en la que la VM es capaz de analizar el comportamiento del programa en ejecución y recompila y optimiza las partes críticas. La recompilación dinámica puede lograr mayor grado de optimización que la compilación tradicional (o estática), ya que puede basar su trabajo en el conocimiento que de primera mano tiene sobre el entorno de ejecución y el conjunto de clases cargadas en memoria. La compilación JIT y la recompilación dinámica permiten a los programas Java aprovechar la velocidad de ejecución del código nativo sin por ello perder la ventaja de la portabilidad en ambos.





La portabilidad es técnicamente difícil de lograr, y el éxito de Java en ese campo ha sido dispar. Aunque es de hecho posible escribir programas para la plataforma Java que actúen de forma correcta en múltiples plataformas de distinta arquitectura, el gran número de estas con pequeños errores o inconsistencias llevan a que a veces se parodie el eslogan de Sun, "Write once, run anywhere" como "Write once, debug everywhere" (o "Escríbelo una vez, ejecútalo en cualquier parte" por "Escríbelo una vez, depúralo en todas partes")

El concepto de independencia de la plataforma de Java cuenta, sin embargo, con un gran éxito en las aplicaciones en el entorno del servidor, como los Servicios Web, los Servlets, los Java Beans, así como en sistemas empotrados basados en OSGi, usando entornos Java empotrados.



## 2.3 - ORACLE

Oracle es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), desarrollado por Oracle Corporation.

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:

### 2.3.1 Gestión de transacciones

Una transacción en un Sistema de Gestión de Bases de Datos (SGBD), es un conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o atómica.

Un SGBD se dice transaccional, si es capaz de mantener la integridad de los datos, haciendo que estas transacciones no puedan finalizar en un estado intermedio. Cuando por alguna causa el sistema debe cancelar la transacción, empieza a deshacer las órdenes ejecutadas hasta dejar la base de datos en su estado inicial (llamado punto de integridad), como si la orden de la transacción nunca se hubiese realizado.

Para esto, el lenguaje de consulta de datos SQL (Structured Query Language), provee los mecanismos para especificar que un conjunto de acciones deben constituir una transacción.

- BEGIN TRAN: Especifica que va a empezar una transacción.
- COMMIT TRAN: Le indica al motor que puede considerar la transacción completada con éxito.
- ROLLBACK TRAN: Indica que se ha alcanzado un fallo y que debe restablecer la base al punto de integridad.

En un sistema ideal, las transacciones deberían garantizar todas las propiedades ACID; en la práctica, a veces alguna de estas propiedades se simplifica o debilita con vistas a obtener un mejor rendimiento.



### 2.3.2 Escalabilidad

En telecomunicaciones y en ingeniería informática, la escalabilidad es la propiedad deseable de un sistema, una red o un proceso, que indica su habilidad para, o bien manejar el crecimiento continuo de trabajo de manera fluida, o bien para estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos. En general, también se podría definir como la capacidad del sistema informático de cambiar su tamaño o configuración para adaptarse a las circunstancias cambiantes. Por ejemplo, una empresa que establece una red de usuarios por Internet, no solamente quiere que su sistema informático tenga capacidad para acoger a los actuales clientes, sino también a los clientes que pueda tener en el futuro y, también, que pueda cambiar su configuración si es necesario.

La escalabilidad como propiedad de los sistemas es generalmente difícil de definir en cualquier caso, en particular es necesario definir los requerimientos específicos para la escalabilidad en esas dimensiones donde se crea que son importantes. Es una edición altamente significativa en sistemas electrónicos, bases de datos, ruteadores y redes. A un sistema cuyo rendimiento es mejorado después de haberle añadido más capacidad hardware, proporcionalmente a la capacidad añadida, se dice que pasa a ser "un sistema escalable"

En concreto, la versión de Oracle que utilizaremos es Oracle 10g Express edition, por ser una versión gratuita de gran potencia que nos permitirá gestionar las BBDD de los clientes. El mantenimiento de las mismas, así como la realización de backups, será un tema importante a tratar en la 3º parte de esta memoria



## **2.4 - POSTFIX**

Postfix es un Agente de Transporte de Correo (MTA) de software libre / código abierto, un programa informático para el enrutamiento y envío de correo electrónico, creado con la intención de que sea una alternativa más rápida, fácil de administrar y segura al ampliamente utilizado Sendmail. Formalmente conocido como VMailer e IBM Secure Mailer, fue originalmente escrito por Wietse Venema durante su estancia en el Thomas J. Watson Research Center de IBM, y continúa siendo desarrollado activamente.

Postfix es el agente de transporte por omisión en diversas distribuciones de Linux y en las últimas versiones del Mac OS X.

En nuestro proyecto, postfix se va a configurar y utilizar simplemente como herramienta de notificación, para enviar mensajes por correo a los diferentes usuarios y validadores.



## **2.5 – ETIQUETAS PERSONALIZADAS (CUSTOMIZED TAGS)**

La tecnología JavaServer Pages Standard Tag Library (JSTL) es un componente de Java EE. Extiende las ya conocidas JavaServer Pages (JSP) proporcionando cuatro librerías de etiquetas (Tag Libraries) con utilidades ampliamente utilizadas en el desarrollo de páginas web dinámicas.

Estas librerías de etiquetas extienden de la especificación de JSP (la cual a su vez extiende de la especificación de Servlet). Su API nos permite además desarrollar nuestras propias librerías de etiquetas.

Su principal fin es la reutilización de código y evitar el uso de scriptlets en los JSP. Se utiliza a lo largo de todo el proyecto para facilitar además la comprensión del código y evitar en la medida de lo posible el uso de “codigo spaghetti”.



## 2.6 – PRINCIPALES FICHEROS DEL SISTEMA

A continuación vamos a indicar los principales ficheros de los que consta el sistema, para de esta manera ayudar en la implementación del mismo:

**Fichero de usuarios:** En este fichero se guardarán todos los usuarios del sistema. Se cargará estáticamente con la aplicación y su contenido se utilizará para validar a los usuarios del mismo cuando se realice el login.

**Fichero de museos:** En este fichero se guardarán todos los museos del sistema. También se cargará estáticamente con la aplicación. Su contenido se utilizará principalmente para poder conectarse a una u otra Base de Datos dependiendo del museo.

**Fichero contador:** Se utilizará para guardar un contador con el último código de pedido utilizado. De esta forma, si tenemos que reiniciar la aplicación el código de pedido que asignamos comenzará a partir del último utilizado en vez de empezar de 0. Esto evita que haya inconsistencias en la BBDD.



### 3 - ANÁLISIS DE REQUISITOS Y DISEÑO DEL PORTAL

Para el diseño del portal web deberemos basarnos en un estándar de Ingeniería de Software que tenga unos sólidos cimientos y una fuerte estabilidad, para este proyecto haremos uso del estándar ESA PSS-05-0, utilizado por la Agencia Espacial Europea, ya que cumple con nuestras expectativas.

Dicho estándar puede consultarse en la página web oficial de la Agencia Espacial Europea, en la siguiente URL de Internet: <http://www.esa.int/esaCP/index.html>

**Aplicando el estándar ESA PSS-05-0 seguiremos los siguientes pasos:**

- 1) Requisitos de Usuario de Capacidad: Estos requisitos representan las capacidades que los usuarios necesitan que tenga el sistema para que pueda resolver sus problemas o cumplir sus objetivos.
- 2) Requisitos de Usuario de Restricción: Estos requisitos representan las restricciones que los usuarios establecen acerca de cómo se solucionarán los problemas o se lograrán los objetivos.
- 3) Diagramas de Casos de Uso: Estos proporcionan uno o más escenarios que indican cómo deberían interactuar el sistema con los usuarios o con otros sistemas para conseguir un objetivo específico.
- 4) Diagrama de Arquitectura: Muestra gráficamente la estructura y relación de las tecnologías software utilizadas en el proyecto.
- 5) Diagramas de Secuencia: Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase.
- 6) Trazabilidad: Capturar las relaciones de implementación y dependencia en el modelo, siguiendo un histórico de los pasos seguidos para desarrollar el software.

### 3.1 – DEFINICIÓN DE REQUISITOS

En este capítulo se exponen los requisitos de usuario y requisitos software de la aplicación que se va a desarrollar y que se utilizará para la simulación.

La definición de requisitos es fundamental para acotar el alcance de la aplicación y asegurar que se conseguirán los resultados deseados.

#### **La definición de los requisitos de un sistema informático se realiza en dos pasos:**

- 1) El primero consiste en entrevistar al cliente, en nuestro caso será con los responsables de los diferentes museos que van a utilizar el sistema, para concretar las ideas que éste pueda tener acerca de “**QUÉ**” tiene que hacer el sistema.
- 2) El segundo paso toma estas ideas y, gracias a la experiencia del analista informático, las transforma en la definición de “**CÓMO**” se construirá el sistema.

En el caso concreto de este proyecto, seguiremos el estándar de Ingeniería del Software: **ESA PSS-05-0**, utilizado por la Agencia Espacial Europea.

Dicho estándar puede consultarse en la página web oficial de la Agencia Espacial Europea, en la siguiente URL de Internet: <http://www.esa.int/esaCP/index.html>





### 3.2 - IDENTIFICACIÓN DE REQUISITOS

La tarea de identificación de requisitos se lleva a cabo mediante entrevistas periódicas con el cliente, en nuestro caso será con los responsables de los diferentes museos que van a utilizar el sistema

Dichas entrevistas son entrevistas abiertas (diálogo fluido y espontáneo) o semi-abiertas (combinación de diálogo abierto con inclusión de algunas preguntas preparadas por el entrevistador de antemano), entre los ingenieros del software y el cliente, tomando nota de las necesidades que éste plantee.

Para que la recogida de requisitos se realice de forma clara, sencilla y estructurada se ha definido una plantilla con las siguientes propiedades:

IDENTIFICADOR:			
<b>Nombre:</b>			
<b>Descripción:</b>			
<b>Tipo Usuario:</b>	<input type="checkbox"/> <i>Anónimo</i>	<input type="checkbox"/> <i>Registrado</i>	<input type="checkbox"/> <i>Administrador</i>
<b>Prioridad:</b>	<input type="checkbox"/> <i>Alta</i>	<input type="checkbox"/> <i>Media</i>	<input type="checkbox"/> <i>Baja</i>
<b>Estabilidad:</b>	<input type="checkbox"/> <i>Si</i>	<input type="checkbox"/> <i>No</i>	
<b>Necesidad:</b>	<input type="checkbox"/> <i>Esencial</i>	<input type="checkbox"/> <i>Opcional</i>	<input type="checkbox"/> <i>Conveniente</i>
<b>Fuente:</b>			

Figura 10: Plantilla para Identificación de Requisitos

A continuación explicamos brevemente el significado de los parámetros que se recogen en la plantilla:

- **Identificador:** Identifica de forma unívoca cada uno de los requisitos. Dicho identificador sigue la siguiente nomenclatura.

**IDENTIFICADOR:** RU+Tipo+Número

Donde:

- **RU:** Son las siglas de “Requisito de Usuario”.
- **Tipo:** Puede tomar los valores “C”, si se trata de un requisito de usuario de Capacidad, y “R” si se trata de un requisito de usuario de Restricción.
- **Número:** Será un número de tres cifras que empezará por 001 y se irá incrementado en una unidad por cada nuevo requisito añadido.



Ejemplos:

- RUC001: Requisito de Usuario de Capacidad, Número 1.
  - RUR010: Requisito de Usuario de Restricción, Número 10.
- **Nombre:** Expresa el nombre del requisito, en pocas palabras un resumen del requisito.
- **Descripción:** Breve comentario textual del requisito.
- **Tipo Usuario:** Nos indica el tipo de usuario para el cual va dirigido el requisito. Puede tomar los siguientes valores:
- “Anónimo”: Usuario visitante del portal, dicho usuario no puede realizar ninguna acción sobre nuestro sistema hasta no haberse logado con algún otro perfil que no sea de anónimo.
  - “Usuario”: En nuestro caso el perfil usuario se refiere a usuario para poder realizar consultas y solicitar préstamos sobre obras de arte de otros museos. Asimismo, podrá consultar los préstamos realizados y el estado de los mismos.
  - “Validador”: El validador posee los privilegios para gestionar todas las solicitudes de préstamo que se realicen al museo sobre el cual tenga privilegios.
- **Prioridad:** Indica la prioridad en el desarrollo del requisito. Puede tomar los valores “Alta”, “Media” o “Baja”.
- **Estabilidad:** Indica la posibilidad de que el requisito cambie a lo largo del desarrollo de la aplicación. Puede tomar los siguientes valores:
- “Si”: Cuando el cliente asegura que no va a ser modificado.
  - “No”: Cuando el requisito puede variar en función de las sucesivas etapas del proyecto.
- **Necesidad:** Indica el nivel de necesidad del requisito dentro del sistema final. Puede tomar los valores “Esencial”, “Opcional” o “Conveniente”.



- “Esencial”: Cuando el cliente no acepte ninguna negociación.
  - “Conveniente”: Cuando el requisito se pueda negociar.
  - “Opcional”: Cuando su implementación puede ser eliminada.
- **Fuente:** Indica el origen a partir del cual se ha obtenido el documento. Cuando se trata de un documento externo se hace referencia a dicho documento.



### 3.3 - REQUISITOS DE USUARIO

En la fase de toma de requisitos de usuario, o fase de definición del problema, se definirán el ámbito y el alcance del sistema, es decir, lo que se espera que haga el sistema.

Los requisitos de usuario son de dos tipos: de capacidad y de restricción.

#### 3.3.1 – Descripción General

##### Perspectiva del Producto:

- El sistema deberá ser capaz de permitir acceso al sistema según el tipo de perfil de usuario que se valide, presentándole además todas las opciones que tenga disponibles en función del mismo.

##### Capacidades Generales:

El sistema ofrecerá diversas funcionalidades que podemos clasificar en función del usuario que va a realizarlas:

- Usuario:
  - Realizar una consulta sobre una obra de arte
  - Solicitar el préstamo de la misma
  - Realizar una consulta sobre el estado de sus solicitudes de préstamo
- Administrador:
  - Validar o denegar una solicitud de préstamo
  - Consultar todas las solicitudes realizadas al museo sobre el que consta como validador
  - Consultar información sobre cualquier usuario del sistema que haya realizado una petición a su museo.

##### Restricciones Generales:

El sistema tendrá que poderse utilizar de manera remota, tanto por los usuarios como por el validador, siendo instalado dicho sistema en un servidor y accediéndose a este mediante su URL.

A su vez, dicho acceso debe realizarse a través de un canal convencional http.



### **Características de los posibles usuarios del sistema:**

A este sistema accederán dos tipos de usuarios:

- Usuario Anónimo: Éste no necesita ninguna capacidad especial para acceder al contenido público del portal, tan sólo conocer la dirección de Internet para acceder desde su navegador. Para un usuario anónimo,
- Usuario Registrado: Éste necesitará conocer los parámetros “Usuario” y “Clave”, para acceder al contenido privado del portal. Una vez dentro podrá realizar todas las acciones correspondientes al perfil usuario.
- Validador: Éste también necesitará conocer los parámetros “Usuario” y “Clave”, para acceder al contenido privado del portal. Una vez dentro podrá realizar todas las acciones correspondientes al perfil validador.

### **Entorno de Operación:**

Todas las operaciones se realizarán de forma remota, ya que dicho portal será instalado en un servidor de Internet.

Las pruebas de funcionamiento se podrán realizar en modo local, teniendo instalado un servidor Glassfish cualquier junto con una BBDD ORACLE.



### 3.3.2 – Requisitos de Usuario de Capacidad

Estos requisitos representan las capacidades que los usuarios necesitan que tenga el sistema para que pueda resolver sus problemas o cumplir sus objetivos.

A continuación se exponen los requisitos de capacidad:

IDENTIFICADOR: RUC001	
<b>Nombre:</b>	<b>Login en el sistema</b>
<b>Descripción:</b>	El sistema debe permitir a un usuario anónimo poder logarse en el mismo. El login se realizará mediante el sistema usuario/contraseña, y será la única manera por la cual el usuario podrá entrar en el mismo..
<b>Tipo Usuario:</b>	<input checked="" type="checkbox"/> <i>Anónimo</i> <input type="checkbox"/> <i>Usuario</i> <input type="checkbox"/> <i>Validador</i>
<b>Prioridad:</b>	<input checked="" type="checkbox"/> <i>Alta</i> <input type="checkbox"/> <i>Media</i> <input type="checkbox"/> <i>Baja</i>
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> <i>Si</i> <input type="checkbox"/> <i>No</i>
<b>Necesidad:</b>	<input checked="" type="checkbox"/> <i>Esencial</i> <input type="checkbox"/> <i>Opcional</i> <input type="checkbox"/> <i>Conveniente</i>
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).

IDENTIFICADOR: RUC002	
<b>Nombre:</b>	<b>Consultar Obras de arte</b>
<b>Descripción:</b>	Debe permitir, introduciendo una clave, poder consultar la información de todas las obras de arte en todos los museos que estén en el sistema, en cuyo título o autor aparezca dicha clave.
<b>Tipo Usuario:</b>	<input type="checkbox"/> <i>Anónimo</i> <input checked="" type="checkbox"/> <i>Usuario</i> <input type="checkbox"/> <i>Validador</i>
<b>Prioridad:</b>	<input checked="" type="checkbox"/> <i>Alta</i> <input type="checkbox"/> <i>Media</i> <input type="checkbox"/> <i>Baja</i>
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> <i>Si</i> <input type="checkbox"/> <i>No</i>
<b>Necesidad:</b>	<input checked="" type="checkbox"/> <i>Esencial</i> <input type="checkbox"/> <i>Opcional</i> <input type="checkbox"/> <i>Conveniente</i>
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).



IDENTIFICADOR: RUC003	
<b>Nombre:</b>	<b>Añadir y eliminar obras de arte del pedido</b>
<b>Descripción:</b>	Posterior a su consulta, el usuario tendrá la posibilidad de añadir a un pedido una determinada obra de arte. También tendrá la posibilidad de eliminar las obras de arte que tenga en su pedido.
<b>Tipo Usuario:</b>	<input type="checkbox"/> <i>Anónimo</i> <input checked="" type="checkbox"/> <i>Usuario</i> <input type="checkbox"/> <i>Validador</i>
<b>Prioridad:</b>	<input checked="" type="checkbox"/> <i>Alta</i> <input type="checkbox"/> <i>Media</i> <input type="checkbox"/> <i>Baja</i>
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> <i>Si</i> <input type="checkbox"/> <i>No</i>
<b>Necesidad:</b>	<input checked="" type="checkbox"/> <i>Esencial</i> <input type="checkbox"/> <i>Opcional</i> <input type="checkbox"/> <i>Conveniente</i>
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).

IDENTIFICADOR: RUC004	
<b>Nombre:</b>	<b>Enviar Solicitud de Pedido</b>
<b>Descripción:</b>	Una vez el usuario haya decidido que obras desea solicitar, el sistema le pedirá que introduzca las fechas inicial y final, para posteriormente insertar la solicitud en la base de datos de cada museo cuya obra de arte se desee solicitar.
<b>Tipo Usuario:</b>	<input type="checkbox"/> <i>Anónimo</i> <input checked="" type="checkbox"/> <i>Usuario</i> <input type="checkbox"/> <i>Validador</i>
<b>Prioridad:</b>	<input checked="" type="checkbox"/> <i>Alta</i> <input type="checkbox"/> <i>Media</i> <input type="checkbox"/> <i>Baja</i>
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> <i>Si</i> <input type="checkbox"/> <i>No</i>
<b>Necesidad:</b>	<input checked="" type="checkbox"/> <i>Esencial</i> <input type="checkbox"/> <i>Opcional</i> <input type="checkbox"/> <i>Conveniente</i>
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).

IDENTIFICADOR: RUC005	
<b>Nombre:</b>	<b>Consultar Pedidos realizados</b>
<b>Descripción:</b>	El usuario tiene la posibilidad de consultar el estado de sus pedidos.
<b>Tipo Usuario:</b>	<input type="checkbox"/> <i>Anónimo</i> <input checked="" type="checkbox"/> <i>Usuario</i> <input type="checkbox"/> <i>Validador</i>
<b>Prioridad:</b>	<input checked="" type="checkbox"/> <i>Alta</i> <input type="checkbox"/> <i>Media</i> <input type="checkbox"/> <i>Baja</i>
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> <i>Si</i> <input type="checkbox"/> <i>No</i>
<b>Necesidad:</b>	<input checked="" type="checkbox"/> <i>Esencial</i> <input type="checkbox"/> <i>Opcional</i> <input type="checkbox"/> <i>Conveniente</i>
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).



IDENTIFICADOR: RUC006	
<b>Nombre:</b>	<b>Consultar Info. Adicional Obras</b>
<b>Descripción:</b>	Permite al usuario pedir información extra acerca de una obra de arte concreta, como la técnica, el tamaño etc...
<b>Tipo Usuario:</b>	<input type="checkbox"/> <i>Anónimo</i> <input checked="" type="checkbox"/> <i>Registrado</i> <input type="checkbox"/> <i>Administrador</i>
<b>Prioridad:</b>	<input checked="" type="checkbox"/> <i>Alta</i> <input type="checkbox"/> <i>Media</i> <input type="checkbox"/> <i>Baja</i>
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> <i>Si</i> <input type="checkbox"/> <i>No</i>
<b>Necesidad:</b>	<input type="checkbox"/> <i>Esencial</i> <input type="checkbox"/> <i>Opcional</i> <input checked="" type="checkbox"/> <i>Conveniente</i>
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).

IDENTIFICADOR: RUC007	
<b>Nombre:</b>	<b>Validar Peticiones</b>
<b>Descripción:</b>	El validador recibirá una lista con todas las peticiones pendientes de validar. Podrá decidir validar o negar la solicitud de préstamo de dicha obra
<b>Tipo Usuario:</b>	<input type="checkbox"/> <i>Anónimo</i> <input type="checkbox"/> <i>Usuario</i> <input checked="" type="checkbox"/> <i>Validador</i>
<b>Prioridad:</b>	<input checked="" type="checkbox"/> <i>Alta</i> <input type="checkbox"/> <i>Media</i> <input type="checkbox"/> <i>Baja</i>
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> <i>Si</i> <input type="checkbox"/> <i>No</i>
<b>Necesidad:</b>	<input checked="" type="checkbox"/> <i>Esencial</i> <input type="checkbox"/> <i>Opcional</i> <input type="checkbox"/> <i>Conveniente</i>
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).

IDENTIFICADOR: RUC008	
<b>Nombre:</b>	<b>Consultar Info Usuarios</b>
<b>Descripción:</b>	Debe permitir al validador consultar la información adicional de cualquier usuario que haya solicitado un préstamo a dicho museo
<b>Tipo Usuario:</b>	<input type="checkbox"/> <i>Anónimo</i> <input type="checkbox"/> <i>Usuario</i> <input checked="" type="checkbox"/> <i>Validador</i>
<b>Prioridad:</b>	<input checked="" type="checkbox"/> <i>Alta</i> <input type="checkbox"/> <i>Media</i> <input type="checkbox"/> <i>Baja</i>
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> <i>Si</i> <input type="checkbox"/> <i>No</i>
<b>Necesidad:</b>	<input checked="" type="checkbox"/> <i>Esencial</i> <input type="checkbox"/> <i>Opcional</i> <input type="checkbox"/> <i>Conveniente</i>
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).





IDENTIFICADOR: RUC009	
<b>Nombre:</b>	<b>Enviar Mail de confirmación</b>
<b>Descripción:</b>	El sistema automáticamente enviará un mail de confirmación o rechazo al usuario cuya solicitud se haya validado o rechazado por el validador.
<b>Tipo Usuario:</b>	<input type="checkbox"/> Anónimo <input type="checkbox"/> Usuario <input checked="" type="checkbox"/> Validador
<b>Prioridad:</b>	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No
<b>Necesidad:</b>	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).

IDENTIFICADOR: RUC010	
<b>Nombre:</b>	<b>Consultar pedidos validados</b>
<b>Descripción:</b>	Muestra una lista con todos los pedidos validados por el validador de un determinado museo
<b>Tipo Usuario:</b>	<input type="checkbox"/> Anónimo <input type="checkbox"/> Usuario <input checked="" type="checkbox"/> Validador
<b>Prioridad:</b>	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No
<b>Necesidad:</b>	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).

IDENTIFICADOR: RUC011	
<b>Nombre:</b>	<b>Cambiar Información de contacto</b>
<b>Descripción:</b>	Permite a Usuarios y Validadores del sistema cambiar su mail y su número de teléfono de contacto. Asimismo automáticamente el sistema enviará los mails de confirmación a la nueva dirección del usuario
<b>Tipo Usuario:</b>	<input type="checkbox"/> Anónimo <input checked="" type="checkbox"/> Usuario <input checked="" type="checkbox"/> Validador
<b>Prioridad:</b>	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No
<b>Necesidad:</b>	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).



IDENTIFICADOR: RUC012	
<b>Nombre:</b>	<b>Desconexion</b>
<b>Descripción:</b>	Permite al usuario desconectarse del sistema
<b>Tipo Usuario:</b>	<input type="checkbox"/> <i>Anónimo</i> <input checked="" type="checkbox"/> <i>Registrado</i> <input checked="" type="checkbox"/> <i>Administrador</i>
<b>Prioridad:</b>	<input type="checkbox"/> <i>Alta</i> <input checked="" type="checkbox"/> <i>Media</i> <input type="checkbox"/> <i>Baja</i>
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> <i>Si</i> <input type="checkbox"/> <i>No</i>
<b>Necesidad:</b>	<input type="checkbox"/> <i>Esencial</i> <input type="checkbox"/> <i>Opcional</i> <input checked="" type="checkbox"/> <i>Conveniente</i>
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).



### 3.3.3 – Requisitos de Usuario de Restricción

Estos requisitos representan las restricciones que los usuarios establecen acerca de cómo se solucionarán los problemas o se lograrán los objetivos.

A continuación se exponen los requisitos de restricción:

IDENTIFICADOR: RUR001	
<b>Nombre:</b>	<b>Servidor Glassfish V2 o superior</b>
<b>Descripción:</b>	La versión del Servidor de aplicaciones debe ser Glassfish 2 o superior para garantizar el correcto funcionamiento del sistema
<b>Prioridad:</b>	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No
<b>Necesidad:</b>	<input type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input checked="" type="checkbox"/> Conveniente
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).

IDENTIFICADOR: RUR002	
<b>Nombre:</b>	<b>Gestor de Bases de Datos ORACLE 10g</b>
<b>Descripción:</b>	La versión del Gestor de Bases de Datos SQL debe ser ORACLE 10g para asegurarse el correcto funcionamiento del sistema desarrollado.
<b>Prioridad:</b>	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No
<b>Necesidad:</b>	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).

IDENTIFICADOR: RUR003	
<b>Nombre:</b>	<b>Intérprete de Java</b>
<b>Descripción:</b>	La versión del Intérprete de Java debe ser J2SE 1.6(Mustang) o superior para asegurarse el correcto funcionamiento del sistema desarrollado.
<b>Prioridad:</b>	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No
<b>Necesidad:</b>	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).



IDENTIFICADOR: RUR004	
<b>Nombre:</b>	<b>Servidor de correo</b>
<b>Descripción:</b>	El servidor de correo utilizado debe ser Postfix v 2.3 o superior para el correcto funcionamiento del sistema.
<b>Prioridad:</b>	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No
<b>Necesidad:</b>	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).

IDENTIFICADOR: RUR005	
<b>Nombre:</b>	<b>Resolución de la Pantalla</b>
<b>Descripción:</b>	La resolución para la cual la visualización de la aplicación Web es óptima es de 1280x800.
<b>Prioridad:</b>	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No
<b>Necesidad:</b>	<input type="checkbox"/> Esencial <input type="checkbox"/> Opcional <input checked="" type="checkbox"/> Conveniente
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).

IDENTIFICADOR: RUR006	
<b>Nombre:</b>	<b>Navegador</b>
<b>Descripción:</b>	El navegador en el cuál se deben visualizar de manera óptima las interfaces del Gestor de Contenidos es para Internet Explorer 6.0.
<b>Prioridad:</b>	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No
<b>Necesidad:</b>	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Opcional <input checked="" type="checkbox"/> Conveniente
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).

IDENTIFICADOR: RUR007	
<b>Nombre:</b>	<b>Entorno Windows</b>
<b>Descripción:</b>	El entorno donde será desarrollado y mantenida la aplicación Web será Windows XP Professional o cualquier sistema Operativo compatible con el mismo y su versión.
<b>Prioridad:</b>	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No
<b>Necesidad:</b>	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Opcional <input type="checkbox"/> Conveniente
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).



<b>IDENTIFICADOR:</b> RUR008	
<b>Nombre:</b>	<b>Librerías JSTL</b>
<b>Descripción:</b>	La versión de librerías JSTL utilizadas debe ser JSTL v1.4 o superior para el correcto funcionamiento del sistema.
<b>Prioridad:</b>	<input checked="" type="checkbox"/> <i>Alta</i> <input type="checkbox"/> <i>Media</i> <input type="checkbox"/> <i>Baja</i>
<b>Estabilidad:</b>	<input checked="" type="checkbox"/> <i>Si</i> <input type="checkbox"/> <i>No</i>
<b>Necesidad:</b>	<input checked="" type="checkbox"/> <i>Esencial</i> <input type="checkbox"/> <i>Opcional</i> <input type="checkbox"/> <i>Conveniente</i>
<b>Fuente:</b>	Responsables de los museos que van a utilizar el sistema (El cliente).

## 4 – DIAGRAMA DE CASOS DE USO GENERAL

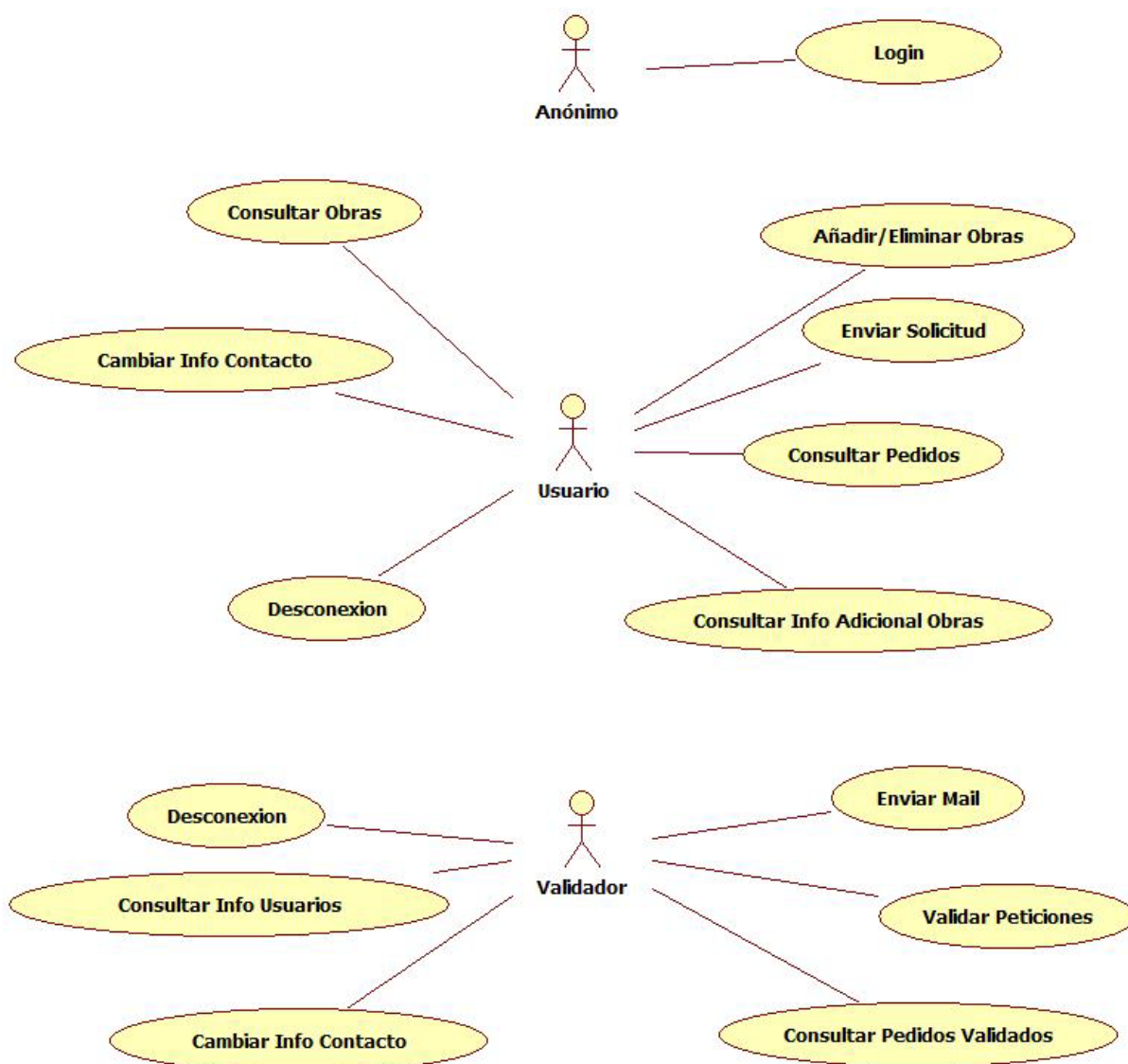


Figura 11: Diagrama de Casos de Uso General

#### 4.1 – Diagrama de casos de uso para el usuario anónimo

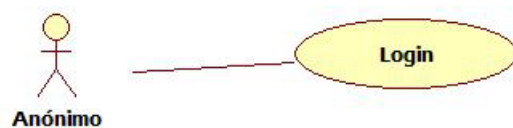


Figura 12: Diagrama de Casos de Uso para el Usuario Anónimo

IDENTIFICADOR: CU001	
<b>Nombre:</b>	<b>Log In</b>
<b>Descripción:</b>	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad “RUC001”. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
<b>Tipo Usuario:</b>	<input checked="" type="checkbox"/> <i>Anónimo</i> <input type="checkbox"/> <i>Registrado</i> <input type="checkbox"/> <i>Validador</i>
<b>Trazabilidad:</b>	RUC001

## 4.2 – Diagrama de casos de uso para el usuario registrado

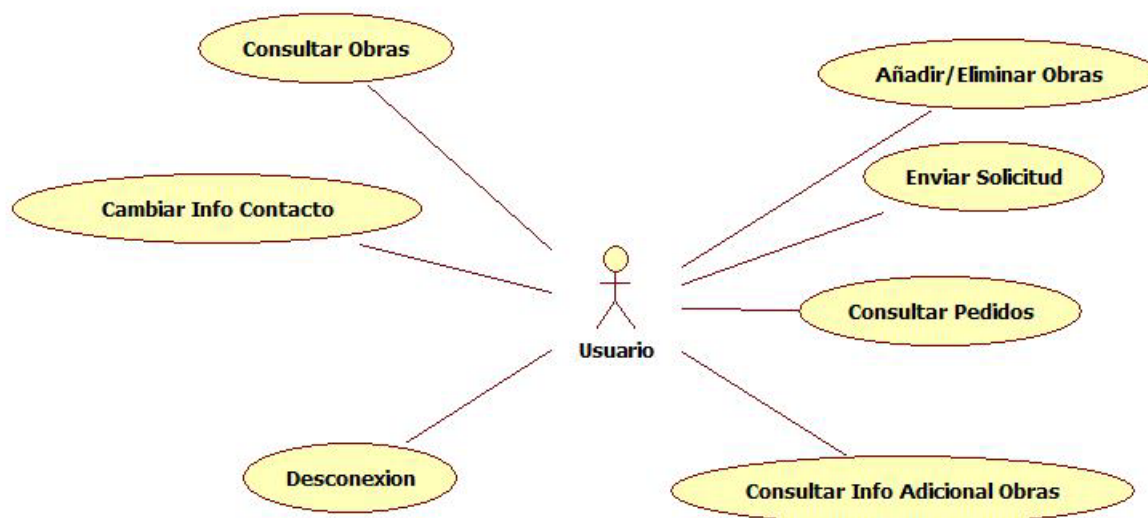


Figura 13: Diagrama de Casos de Uso para el Usuario Registrado

IDENTIFICADOR: CU002	
<b>Nombre:</b>	<b>Consultar Obras</b>
<b>Descripción:</b>	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad “RUC002”. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
<b>Tipo Usuario:</b>	<input type="checkbox"/> Anónimo <input checked="" type="checkbox"/> Registrado <input type="checkbox"/> Validador
<b>Trazabilidad:</b>	RUC002

IDENTIFICADOR: CU003	
<b>Nombre:</b>	<b>Añadir Obras Pedido</b>
<b>Descripción:</b>	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad “RUC003”. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
<b>Tipo Usuario:</b>	<input type="checkbox"/> Anónimo <input checked="" type="checkbox"/> Registrado <input type="checkbox"/> Validador
<b>Trazabilidad:</b>	RUC003





IDENTIFICADOR: CU004	
<b>Nombre:</b>	<b>Enviar Solicitud</b>
<b>Descripción:</b>	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad “RUC004”. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
<b>Tipo Usuario:</b>	<input type="checkbox"/> <i>Anónimo</i> <input checked="" type="checkbox"/> <i>Registrado</i> <input type="checkbox"/> <i>Validador</i>
<b>Trazabilidad:</b>	RUC004

IDENTIFICADOR: CU005	
<b>Nombre:</b>	<b>Consultar Pedidos</b>
<b>Descripción:</b>	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad “RUC005”. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
<b>Tipo Usuario:</b>	<input type="checkbox"/> <i>Anónimo</i> <input checked="" type="checkbox"/> <i>Registrado</i> <input type="checkbox"/> <i>Validador</i>
<b>Trazabilidad:</b>	RUC005

IDENTIFICADOR: CU006	
<b>Nombre:</b>	<b>Consultar Info_adicional_Obras</b>
<b>Descripción:</b>	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad “RUC006”. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
<b>Tipo Usuario:</b>	<input type="checkbox"/> <i>Anónimo</i> <input checked="" type="checkbox"/> <i>Registrado</i> <input type="checkbox"/> <i>Validador</i>
<b>Trazabilidad:</b>	RUC006

IDENTIFICADOR: CU011	
<b>Nombre:</b>	<b>Cambiar Información de Contacto</b>
<b>Descripción:</b>	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad “RUC011”. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
<b>Tipo Usuario:</b>	<input type="checkbox"/> Anónimo <input checked="" type="checkbox"/> Registrado <input type="checkbox"/> Validador
<b>Trazabilidad:</b>	RUC011

IDENTIFICADOR: CU012	
<b>Nombre:</b>	<b>Desconexión</b>
<b>Descripción:</b>	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad “RUC012”. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
<b>Tipo Usuario:</b>	<input type="checkbox"/> Anónimo <input checked="" type="checkbox"/> Registrado <input type="checkbox"/> Validador
<b>Trazabilidad:</b>	RUC012

#### 4.3 – Diagrama de casos de uso para el validador

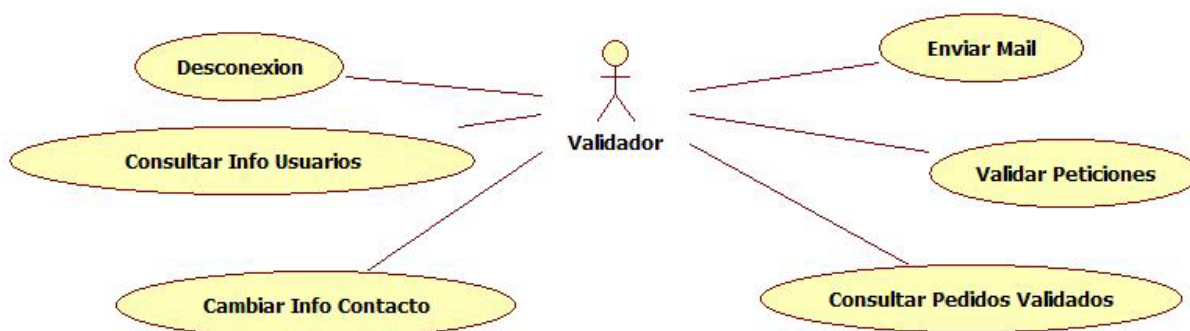


Figura 14: Diagrama de Casos de Uso para el Validador

IDENTIFICADOR: CU007	
<b>Nombre:</b>	<b>Validar Peticiones</b>
<b>Descripción:</b>	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad “RUC007”. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
<b>Tipo Usuario:</b>	<input type="checkbox"/> Anónimo <input type="checkbox"/> Registrado <input checked="" type="checkbox"/> Validador
<b>Trazabilidad:</b>	RUC007



IDENTIFICADOR: CU008	
<b>Nombre:</b>	<b>Consultar Información Usuarios</b>
<b>Descripción:</b>	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad “RUC008”. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
<b>Tipo Usuario:</b>	<input type="checkbox"/> <i>Anónimo</i> <input type="checkbox"/> <i>Registrado</i> <input checked="" type="checkbox"/> <i>Validador</i>
<b>Trazabilidad:</b>	RUC008

IDENTIFICADOR: CU009	
<b>Nombre:</b>	<b>Enviar Mail de Confirmación</b>
<b>Descripción:</b>	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad “RUC009” como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
<b>Tipo Usuario:</b>	<input type="checkbox"/> <i>Anónimo</i> <input type="checkbox"/> <i>Registrado</i> <input checked="" type="checkbox"/> <i>Validador</i>
<b>Trazabilidad:</b>	RUC009

IDENTIFICADOR: CU0010	
<b>Nombre:</b>	<b>Consultar pedidos validados</b>
<b>Descripción:</b>	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad “RUC0010” como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
<b>Tipo Usuario:</b>	<input type="checkbox"/> <i>Anónimo</i> <input type="checkbox"/> <i>Registrado</i> <input checked="" type="checkbox"/> <i>Validador</i>
<b>Trazabilidad:</b>	RUC010

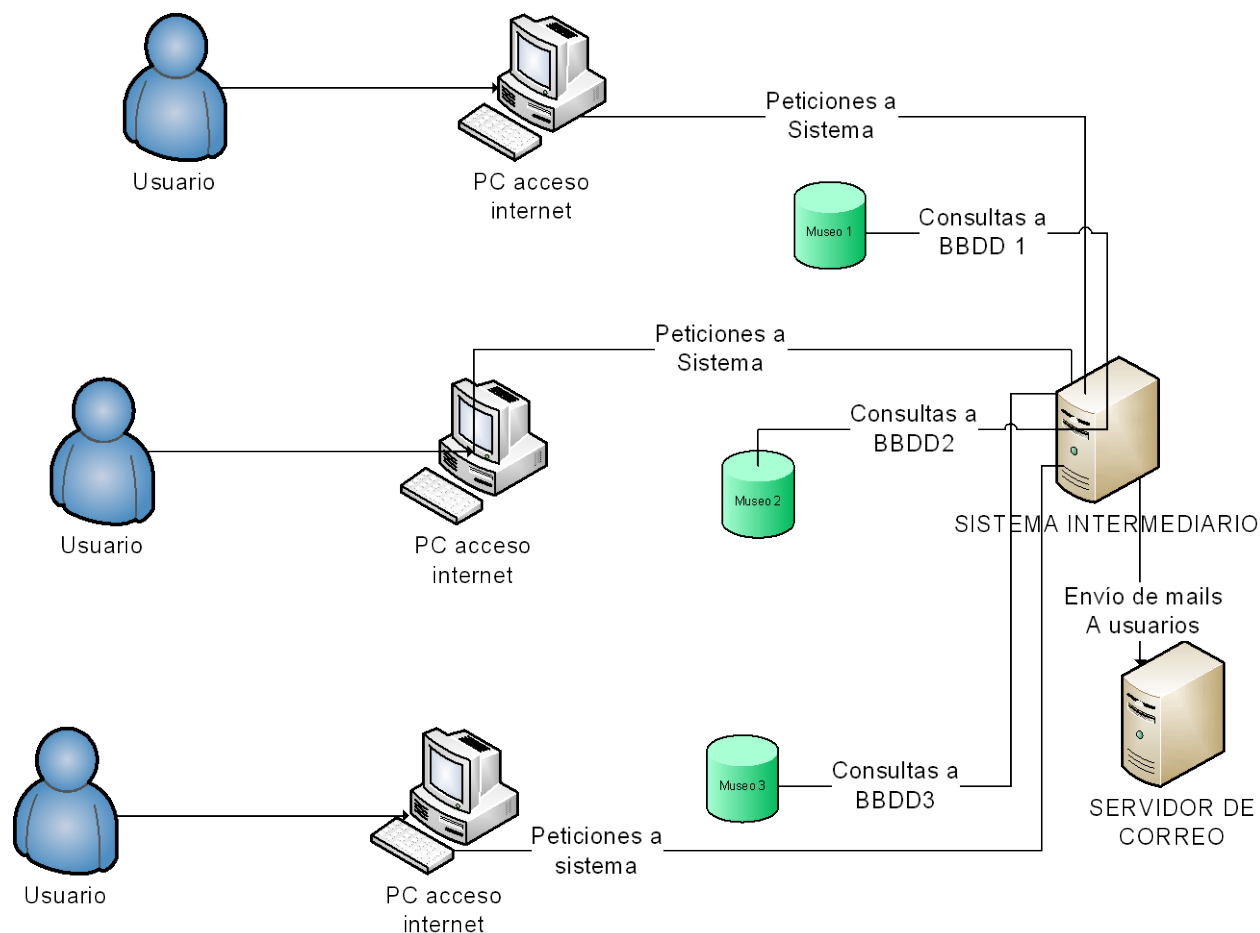


IDENTIFICADOR: CU011	
<b>Nombre:</b>	<b>Cambiar Información de Contacto</b>
<b>Descripción:</b>	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad “RUC011”. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
<b>Tipo Usuario:</b>	<input type="checkbox"/> <i>Anónimo</i> <input type="checkbox"/> <i>Registrado</i> <input checked="" type="checkbox"/> <i>Validador</i>
<b>Trazabilidad:</b>	RUC011

IDENTIFICADOR: CU012	
<b>Nombre:</b>	<b>Desconexion</b>
<b>Descripción:</b>	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad “RUC012”. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
<b>Tipo Usuario:</b>	<input type="checkbox"/> <i>Anónimo</i> <input type="checkbox"/> <i>Registrado</i> <input checked="" type="checkbox"/> <i>Validador</i>
<b>Trazabilidad:</b>	RUC012

## 5 - DIAGRAMA DE ARQUITECTURA

A continuación vamos a indicar la arquitectura general del sistema, y además vamos a indicar mediante un ejemplo como funcionaría dicha arquitectura cuando se realiza una consulta sobre el Sistema Intermediario



Cuando el usuario realiza una consulta sobre el sistema, dicho sistema accede a las bases de datos remotas de todos los museos para devolver al usuario una respuesta. Asimismo, si se configura para ello, cuando un pedido es validado o negado a un usuario, se le envía un mail indicando la aceptación o negación del mismo



Nuestro sistema es una aplicación web cuya función principal será la consulta e inserción de diferentes registros en varias bases de datos remotas. Por lo cual, lo más importante a la hora de realizar dichas acciones es conocer la dirección ip y el SID de dicha base de datos. Las bases de datos pertenecerán a los diferentes museos que estén registrados en el mismo, por lo cual el sistema intermediario no deberá realizar ningún tipo de labor de mantenimiento, pero si será necesario que cada base de datos remota disponga de dos tablas que deben llamarse con el mismo nombre y que tendrán la siguiente información:

### 5.1 Modelo de Datos

Básicamente , todo el sistema ha sido diseñado para trabajar con dos únicas tablas de datos:

Tabla Obras de arte:

En esta tabla irá la información principal sobre las obras de arte. Entre ellas, el ID de la obra, su título, autor, fecha, técnica, medidas, nombre del museo propietario, valor del seguro... Además indicará en caso de estar prestada, la fecha de inicio y fin de dicha obra.

A continuación indicamos el Script de creación de la tabla OBRAS\_ARTE:

```
Create table OBRAS_ARTE
(ID_OBRA char(50) PRIMARY KEY,
TITULO char(50),
FECHA date,
AUTOR char(50),
TECNICA char(25),
MEDIDAS char(50),
NOMBRE_PRESTADOR char(50),
VALOR_SEGURO number(15),
OBSERVACIONES_SEGURO char(100),
DIRECCION_RECOGIDA char(50),
DIRECCION_DEVOLUCION char(50),
OBSERVACIONES_TRANSPORTE char(100),
CORREO char(1),
FECHA_INICIO Date,
FECHA_FIN Date
)
```

Tabla Solicitud de préstamos:

En esta tabla irá información sobre las solicitudes de préstamos realizados a un determinado museo, incluyendo el id de la obra pedida, el id del pedido, el usuario que ha realizado la petición, la fecha de inicio y de fin del mismo, y si está o no validada la solicitud de préstamo.

Sobre este último campo cabe indicar que:

NO: Indica que la solicitud no ha sido aún validada

SI: Indica que la solicitud ha sido validada satisfactoriamente

FF: Indica que la solicitud ha sido validada INSatisfactoriamente

Create table SOL\_PRESTAMOS

```
(ID_OBRA char(50),
ID_PEDIDO CHAR(20),
```

```

ID_USUARIO char(10),
FECHA_INICIO Date,
FECHA_FIN Date,
VALIDADO char(2)
)

```

## 5.2 Funcionamiento general del Sistema

A continuación indicaremos mediante un ejemplo, el funcionamiento general de la arquitectura del sistema.

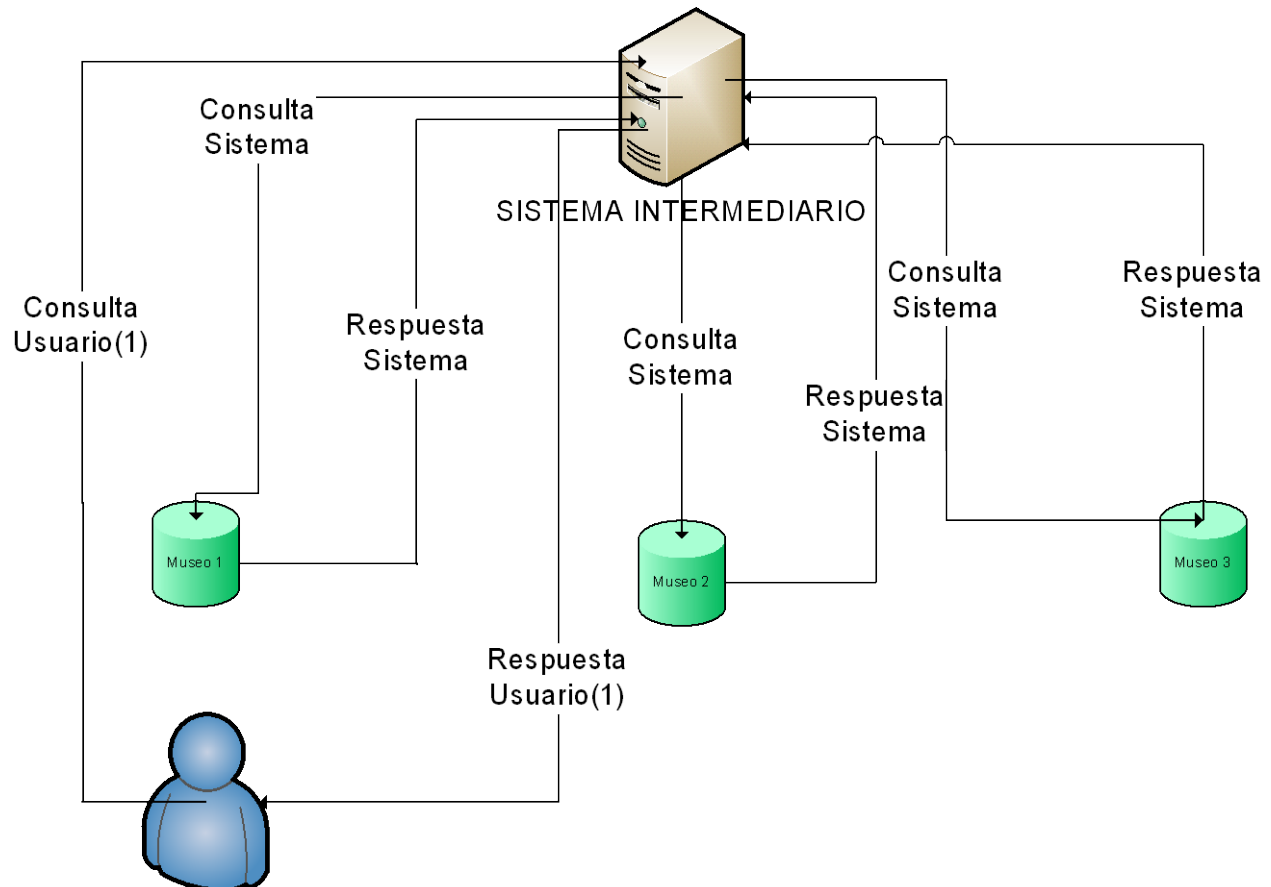


Figura 15: Consulta de una Obra

Cuando un usuario\_registrado hace una consulta al sistema intermedio, este replica dicha consulta a todas las bases de datos remotas de todos los museos, para intentar encontrar el patrón pedido por el usuario.

Cuando se realiza la consulta, el sistema busca tanto por nombre, como por autor de la obra y devuelve todas las obras que contengan el patrón buscado. A continuación, el sistema intermedio muestra la información consultada al usuario.

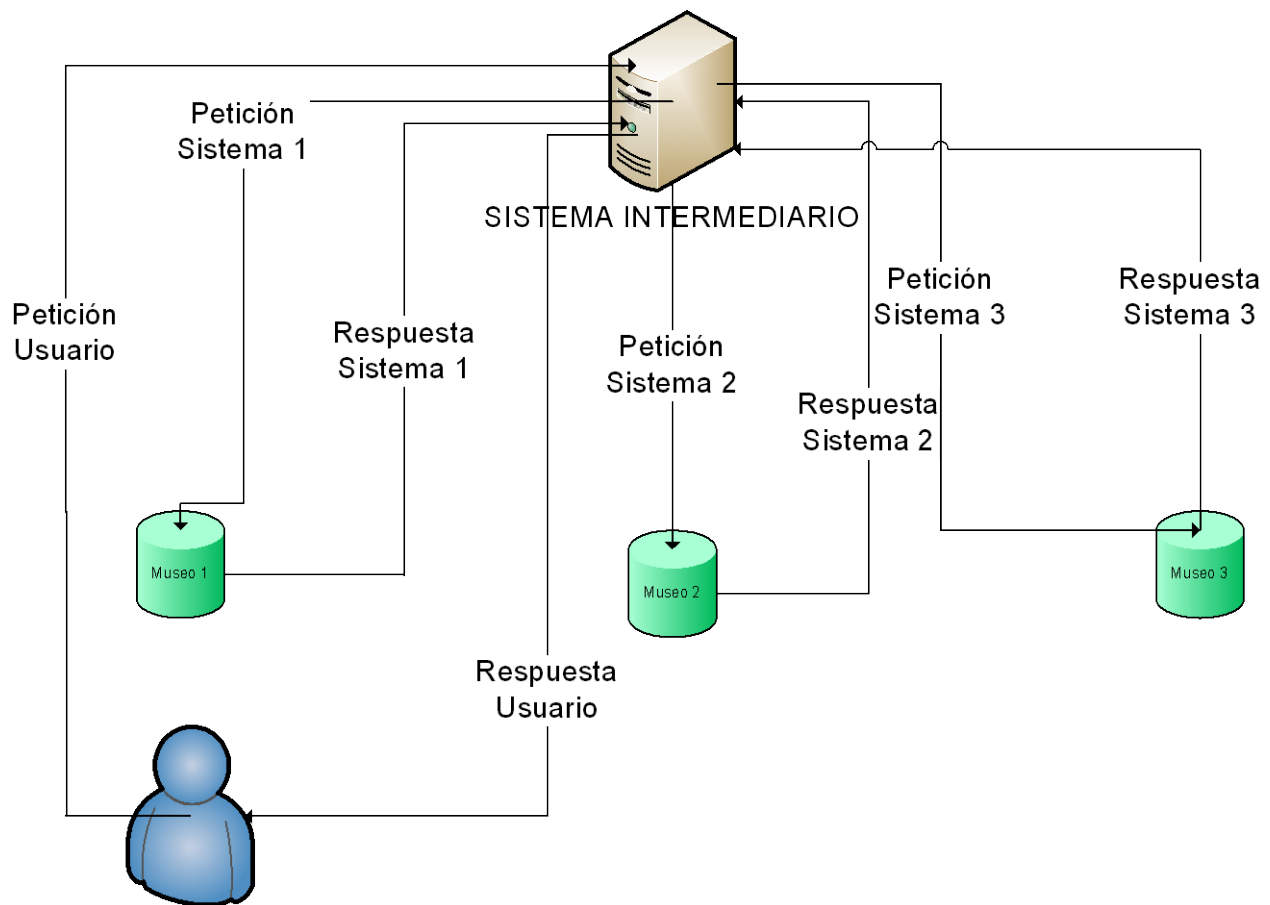


Figura 16: Solicitud de préstamo de un Usuario

Cuando un usuario realiza una petición (que puede tener varias obras de arte a diferentes museos distintos), el Sistema Intermediario se encargará de dividir la petición en varias peticiones distintas, asignando a todas el mismo código de petición, para a continuación insertarlas en la tabla SOL\_PRESTAMOS de cada museo.

Finalmente, el validador de cada museo utilizando la aplicación, insertará en la tabla SOL\_PRESTAMOS por cada solicitud recibida para cada obra, si acepta o rechaza dicha petición.

Debemos además indicar, que el sistema está configurado para enviar un mail de confirmación o negación al usuario que realizó la solicitud cuando su pedido se valida o se deniega. Si bien se podría utilizar cualquier servidor, nosotros recomendamos el servidor de correo POSTFIX para Linux, puesto que es gratuito y muy potente. En caso de utilizar Windows, se ha elegido para su uso el servidor CMDMail





Por último indicar que si bien las BBDD son remotas y deberá ser un administrador de BBDD externo el que las configure, es absolutamente necesario tener disparadores(triggers) sobre la tabla SOL\_PRESTAMOS, de tal forma que cuando el tiempo de préstamo haya terminado, la BBDD borre la obra de dicha tabla, y vuelva a dejar vacías las fechas de inicio y fin en la tabla OBRAS\_ARTE



### **5.3 Arquitectura del sistema**

El principal enfoque a la hora de diseñar el proyecto, es fomentar lo máximo posible la reutilización de código, y la facilidad del mismo para adaptarse a cualquier tipo de cambio o nueva adecuación por parte del cliente.

Para ello hemos utilizado principalmente dos patrones:

- Modelo Vista Controlador(MVC).
- Patrón Bean

### 5.3.1 Modelo Vista Controlador



- **Modelo:** Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permitiendo comparar un número de unidades negativo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos o importes en un carrito de la compra.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Muchos sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos: en MVC corresponde al modelo. La unión entre capa de presentación y capa de negocio conocido en el paradigma de la Programación por capas representaría la integración entre Vista y su correspondiente Controlador de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio de capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo ya que ambos cumplen ciclos de vida muy distintos entre si. Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo



estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.

4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

### **5.3.2 Utilización del Modelo Vista Controlador en el proyecto**

Las principales razones por las que se ha utilizado este patrón en el proyecto han sido las siguientes:

- 1) Es un patrón muy utilizado en la creación de aplicaciones Web
- 2) Permite dividir y estructurar la misma en pequeños módulos gestionados por el controlador.
- 3) Es muy sencillo realizar cualquier tipo de modificación o adecuación al proyecto por la razón anteriormente comentada. Si quisiéramos añadir nueva funcionalidad, únicamente deberíamos diseñarla e incluir una llamada en el controlador, haciendo que cualquier futura modificación en la aplicación sea mucho menos costosa de implementar.
- 4) Por último indicar que el patrón Bean complementa perfectamente a este patrón.



### 5.3.3 Patrón Bean

Un Bean es un componente software que tiene la particularidad de ser reutilizable y así evitar la tediosa tarea de programar los distintos componentes uno a uno. Se puede decir que existen con la finalidad de ahorrarnos tiempo al programar. Es el caso de la mayoría de componentes que manejan los editores visuales más comunes. Los que hayan utilizado Visual Studio, Eclipse o Delphi por ejemplo, ya estarán familiarizados con ellos. Bean viene del inglés "grano de café", y su idea principal es que se reaprovechan "granos de café" que ya han sido creados por otros desarrolladores. Sabiendo esto, un Bean puede representar desde un botón, un grid de resultados, un panel contenedor o un simple campo de texto, hasta otras soluciones mucho más complejas como conexiones a bases de datos, etc.

Debe cumplir los siguientes criterios:

- implementar Serializable.
- tener todos sus atributos privados (private).
- tener métodos set() y get() públicos de los atributos privados que nos interese.
- tener un constructor público por defecto.

### 5.3.4 Utilización del Patrón Bean en el proyecto

Las principales razones por las que se ha utilizado este patron en el proyecto han sido las siguientes:

- 1) Facilita la implementación al encapsular en un objeto la información extraída de la BBDD
- 2) Facilita la reutilización de código
- 3) Complementa el patrón vista-controlador



## 6 - DIAGRAMAS DE SECUENCIA

El diagrama de secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema.

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase.

Mientras que el diagrama de casos de uso permite el modelado de una vista del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos.

Típicamente uno examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si tienes modelada la descripción de cada caso de uso como una secuencia de varios pasos, entonces puedes "caminar sobre" esos pasos para descubrir qué objetos son necesarios para que se puedan seguir los pasos.

Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como vectores horizontales.

Los mensajes se dibujan cronológicamente desde la parte superior del diagrama a la parte inferior; la distribución horizontal de los objetos es arbitraria.

Durante el análisis inicial, el modelador típicamente coloca el nombre de un mensaje en la línea del mensaje. Más tarde, durante el diseño, el nombre es reemplazado con el nombre del método que está siendo llamado por un objeto en el otro. El método llamado, o invocado, pertenece a la definición de la clase instanciada por el objeto en la recepción final del mensaje.

### 6.01 – Diagrama de Secuencia “DS001” Para el Caso de Uso “CU001”

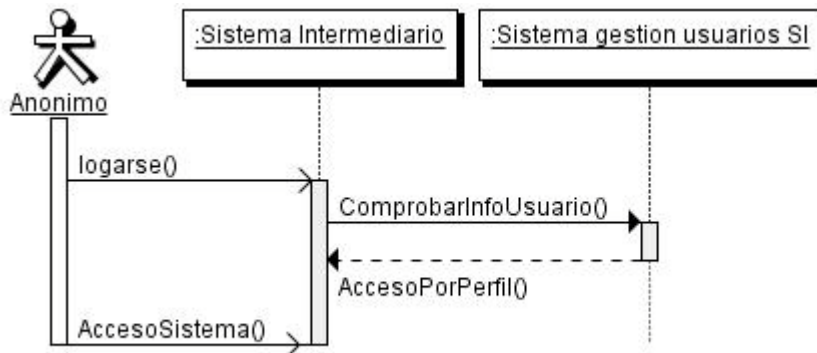


Figura 17: Diagrama de Secuencia “DS001”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU001” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “usuario anónimo” se logea en nuestro sistema.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El Usuario Anónimo introduce los datos en el Sistema Intermediario, estos datos son el DNI y su clave personal de acceso. La clave personal de acceso aparecerá oculta a ojos de cualquiera que pueda ver físicamente el sistema como forma adicional de seguridad.
- 2) El Sistema Intermediario, que previamente habrá cargado en memoria la Gestión de Usuarios (Todos los usuarios que tienen acceso al sistema), y se encargará de consultar la información referente al DNI y clave Introducida como parámetro
- 3) Se devolverá una respuesta en función del Tipo de Perfil con el que aparezca dicho usuario en nuestro sistema, o un mensaje de error en caso de no figurar el DNI/clave introducida.
- 4) El Usuario tendrá acceso a las diferentes opciones según el perfil con el que aparezca en el Sistema de Gestión de Usuarios del Sistema Intermediario.

## 6.02 - Diagrama de Secuencia “DS002” Para el Caso de Uso “CU002”

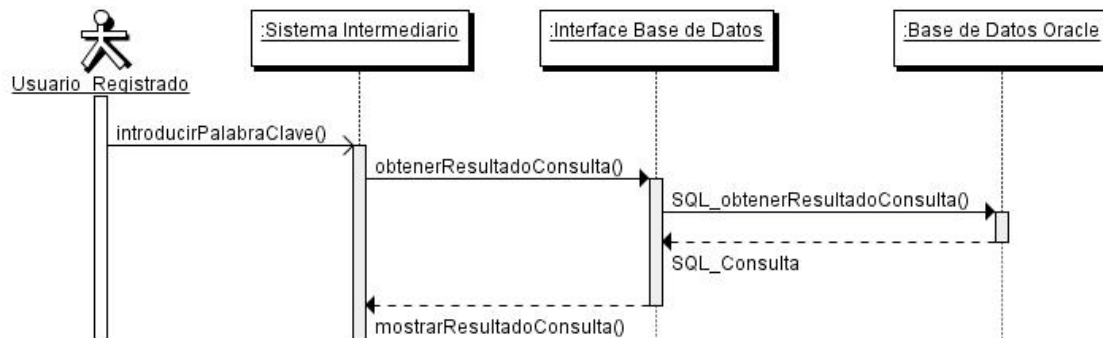


Figura 18: Diagrama de Secuencia “DS002”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU002” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “usuario registrado” realiza la consulta de una obra de arte al sistema.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El Usuario Registrado introduce la palabra clave sobre la obra que desea consultar en el Sistema Intermediario
- 2) El Sistema Intermediario pide al componente denominado Interface Base de Datos que obtenga la información sobre la obra cuya clave ha introducido el usuario.
- 3) El componente Interface Base de Datos a su vez realiza esa petición al Gestor de Bases de Datos Oracle esta vez en lenguaje SQL, para buscar en ésta, los datos sobre la obra cuya clave ha introducido el usuario.
- 4) El Gestor de Bases de Datos Oracle devuelve los datos pedidos en lenguaje SQL al componente Interface Base de Datos.
- 5) El componente “Interface Base de Datos” interpreta el lenguaje SQL y responde al Sistema Intermediario con el resultado que deseaba, para finalmente este mostrarlo por pantalla.



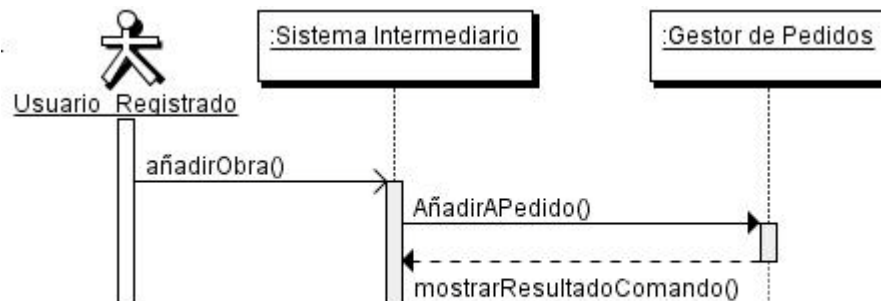
**6.03a - Diagrama de Secuencia “DS003a” Para el Caso de Uso “CU003”**

Figura 19: Diagrama de Secuencia “DS003a”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU003” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “usuario registrado” Añade una obra de arte de a pedido.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El Usuario Registrado marca todas las obras de arte que habiendo consultado previamente, desee añadir al pedido
- 2) El Sistema Intermediario añadirá la o las obras que el usuario ha seleccionado a su pedido.
- 3) Finalmente se mostrará el resultado, es decir, el pedido del Usuario Registrado tras haberse añadido las nuevas obras.

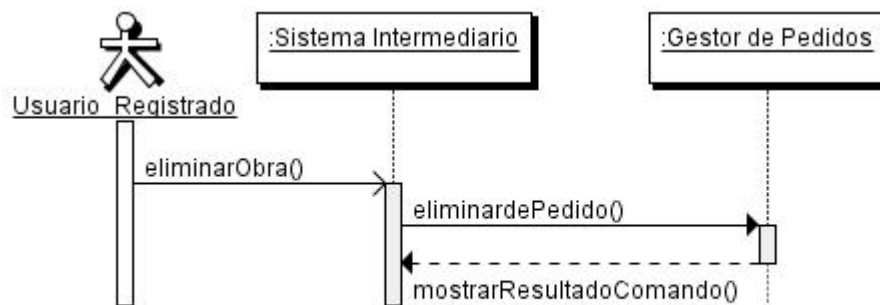
**6.03b - Diagrama de Secuencia “DS003b” Para el Caso de Uso “CU003”**

Figura 20: Diagrama de Secuencia “DS003b”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU003” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “usuario registrado” Elimina una obra de arte de su pedido.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 4) El Usuario Registrado marca todas las obras de arte que habiendo sido añadidas al pedido previamente, desee eliminar del mismo
- 5) El Sistema Intermediario eliminará la o las obras que el usuario ha seleccionado de su pedido.
- 6) Finalmente se mostrará el resultado, es decir, el pedido del Usuario Registrado tras haberse eliminado las nuevas obras.

#### 6.04 - Diagrama de Secuencia “DS004” Para el Caso de Uso “CU004”

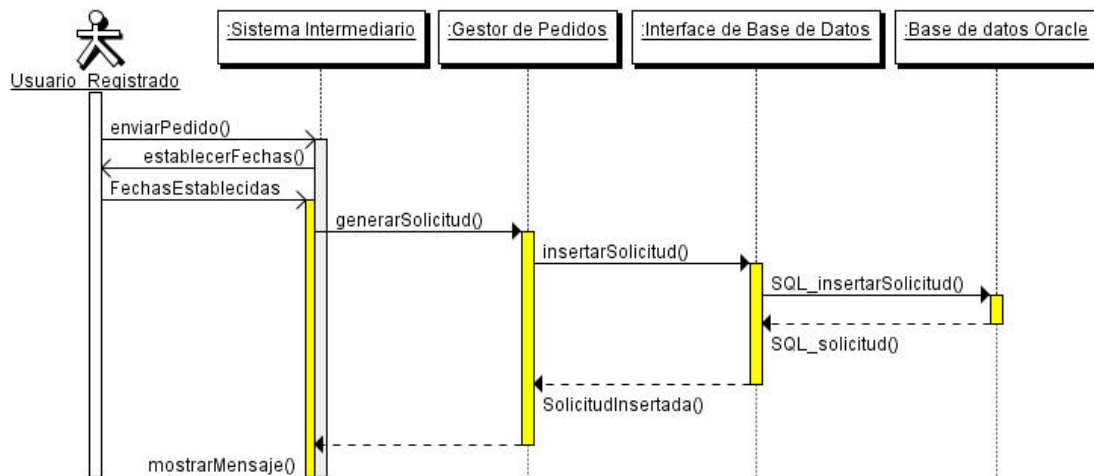


Figura 21: Diagrama de Secuencia “DS004”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU004” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “usuario registrado” realiza la solicitud del pedido.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El Usuario Registrado pulsa la opción para enviar el pedido con las obras previamente seleccionadas.
- 2) El Sistema Intermediario solicita al usuario que introduzca el periodo de inicio y fin de dicho préstamo. El periodo de inicio será por lo menos diez días naturales desde el día actual.
- 3) El usuario introduce las fechas para cada una de las obras solicitadas y pulsa enviar.
- 4) El Sistema se encarga de generar la solicitud correspondiente para cada una de las obras de arte solicitadas, y pasa dicha información al Interface Base de Datos.
- 5) El componente Interface Base de Datos a su vez, se conectará a la base de datos correspondiente para cada una de las obras de arte que aparezcan en el pedido, e irá llamando al Gestor de Bases de Datos para realizar las acciones oportunas.
- 6) El Gestor de Bases de Datos realizará las inserciones de las solicitudes de préstamo en la tabla SOL\_PRESTAMOS de cada uno de los museos, y devolverá un OK o un KO en caso de haber algún error.



- 7) El componente Interface Base de Datos comunicará a su vez dicho OK o KO al Sistema Intermediario.
- 8) En caso de haber ido todo correctamente, el Sistema Intermediario muestra al usuario un mensaje de que su solicitud se ha enviado correctamente. En caso contrario muestra error.

### 6.05 - Diagrama de Secuencia “DS005” Para el Caso de Uso “CU005”

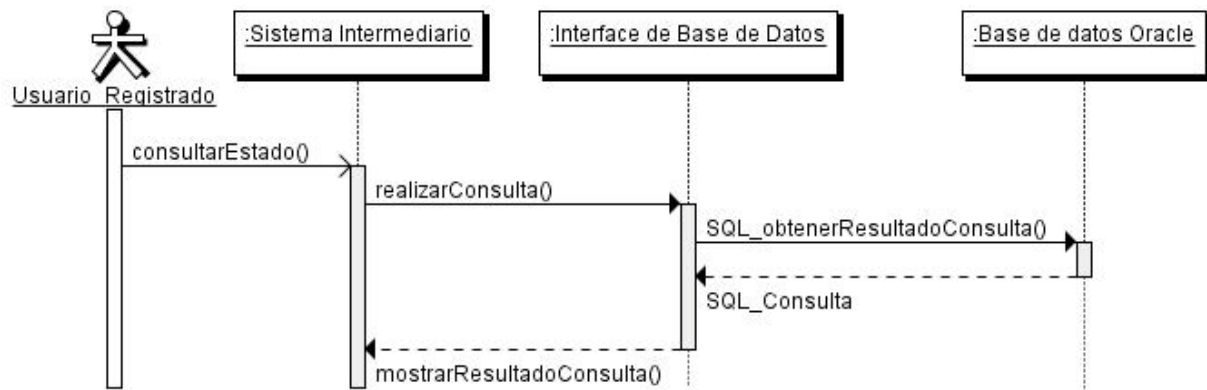


Figura 22: Diagrama de Secuencia “DS005”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU005” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “usuario registrado” consulta los pedidos realizados.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El Usuario Registrado selecciona en el menú lateral la opción “Consultar Pedidos Realizados” en el Sistema Intermediario
- 2) El Sistema Intermediario pide al componente denominado Interface Base de Datos que obtenga la información que ha solicitado el usuario.
- 3) El componente Interface Base de Datos a su vez realiza esa petición al Gestor de Bases de Datos Oracle esta vez en lenguaje SQL. Se consultará en la tabla SOL\_PRESTAMOS de todos los museos en el sistema para ver el estado de los pedidos del Usuario Registrado.
- 4) El Gestor de Bases de Datos Oracle devuelve los datos pedidos en lenguaje SQL al componente Interface Base de Datos.
- 5) El componente “Interface Base de Datos” interpreta el lenguaje SQL y responde al Sistema Intermediario con el resultado que deseaba, para finalmente este mostrarlo por pantalla.

## 6.06 - Diagrama de Secuencia “DS006” Para el Caso de Uso “CU006”

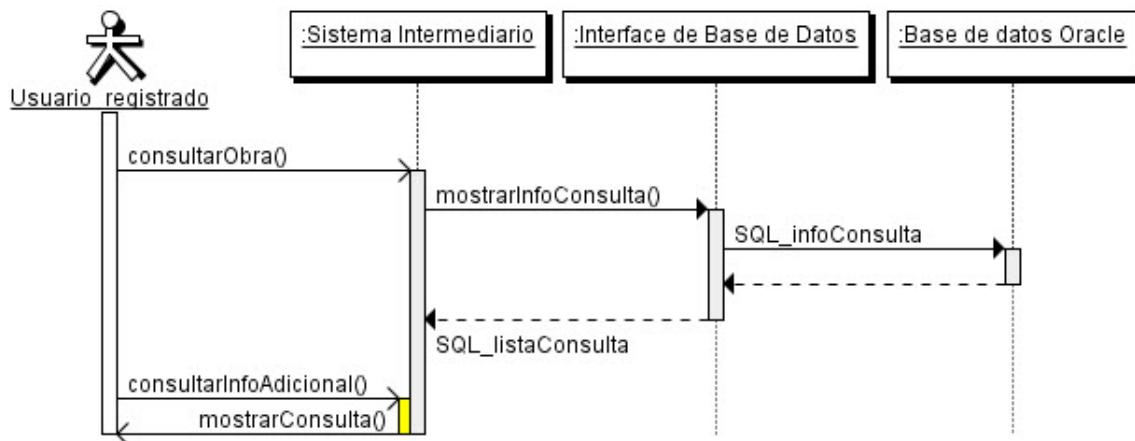


Figura 23: Diagrama de Secuencia “DS006”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU006” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “usuario registrado” consulta información adicional de una obra.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El Usuario Registrado selecciona en el menú lateral la opción “Consultar Obra” en el Sistema Intermediario
- 2) El Sistema Intermediario pide al componente denominado Interface Base de Datos que obtenga la información que ha solicitado el usuario.
- 3) El componente Interface Base de Datos a su vez realiza esa petición al Gestor de Bases de Datos Oracle esta vez en lenguaje SQL. Se consultará en la tabla OBRAS\_ARTE de todos los museos en el sistema para ver la información solicitada.
- 4) El Gestor de Bases de Datos Oracle devuelve los datos pedidos en lenguaje SQL al componente Interface Base de Datos.
- 5) El componente Interface Base de Datos interpreta el lenguaje SQL y responde al Sistema Intermediario con el resultado que deseaba, para este mostrarlo por pantalla.
- 6) A continuación el usuario selecciona el id\_obra cuya información desea ver ampliada y envía la petición al Sistema Intermediario
- 7) El Sistema Intermediario muestra al usuario la información que desea ver.

### 6.07a - Diagrama de Secuencia “DS007a” Para el Caso de Uso “CU007”

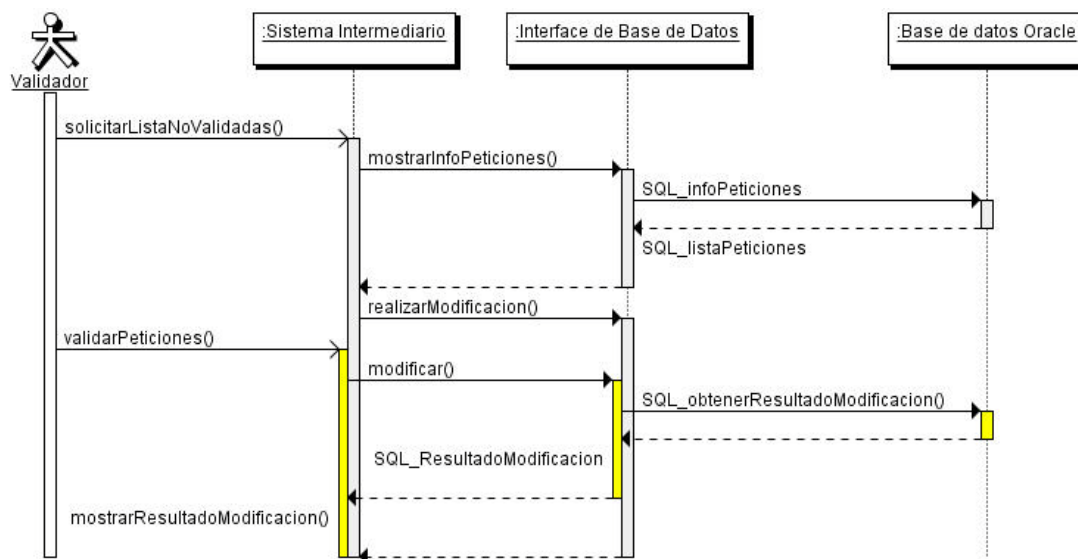


Figura 24: Diagrama de Secuencia “DS007a”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU007” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “Validador” valida una solicitud de pedido.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El Validador selecciona en el menú lateral la opción “Validar Peticiones” en el Sistema Intermediario
- 2) El Sistema Intermediario pide al componente denominado Interface Base de Datos que obtenga un listado de todas las peticiones que han llegado al museo que aún no han sido validadas.
- 3) El componente Interface Base de Datos a su vez realiza esa petición al Gestor de Bases de Datos Oracle esta vez en lenguaje SQL. Se consultará en la tabla SOL\_PRESTAMOS del museo al que pertenece el validador para devolver todos los pedidos que no hayan sido validados por el mismo.
- 4) El Gestor de Bases de Datos Oracle devuelve los datos pedidos en lenguaje SQL al componente Interface Base de Datos.



- 5) El componente “Interface Base de Datos” interpreta el lenguaje SQL y responde al Sistema Intermediario con el resultado que deseaba, para finalmente este mostrarlo por pantalla.
- 6) El Validador selecciona todas aquellas peticiones que desee validar con un OK y pulsa “Validar”
- 7) El Sistema Intermediario pide al componente denominado Interface Base de Datos que realice los cambios indicados por el validador en la BBDD.
- 8) El componente Interface Base de Datos a su vez realiza esa petición al Gestor de Bases de Datos Oracle esta vez en lenguaje SQL. Se insertará en la tabla SOL\_PRESTAMOS del museo al que pertenece el validador para poner el campo Validado a SI
- 9) El Gestor de Bases de Datos Oracle devuelve OK si todo ha ido correctamente al Interface Base de Datos.
- 10) El componente “Interface Base de Datos” interpreta el lenguaje SQL y responde al Sistema Intermediario con un ok si todo ha ido correctamente, y este muestra al Validador un mensaje en pantalla.



### 6.07b - Diagrama de Secuencia “DS007b” Para el Caso de Uso “CU007”

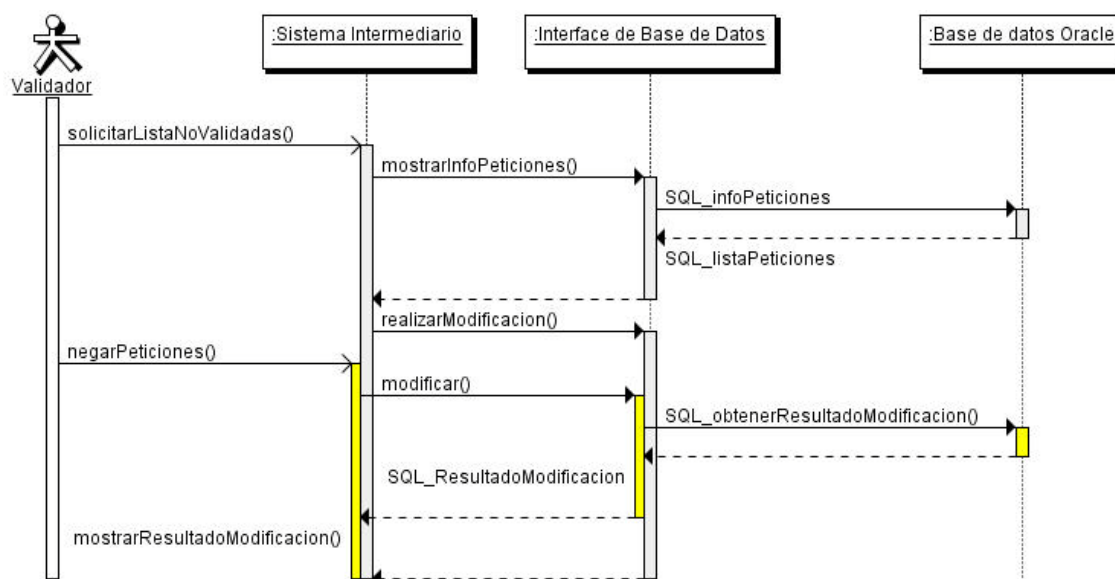


Figura 25: Diagrama de Secuencia “DS007b”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU007” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “Validador” niega una solicitud de pedido.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El Validador selecciona en el menú lateral la opción “Validar Peticiones” en el Sistema Intermediario
- 2) El Sistema Intermediario pide al componente denominado Interface Base de Datos que obtenga un listado de todas las peticiones que han llegado al museo que aún no han sido validadas.
- 3) El componente Interface Base de Datos a su vez realiza esa petición al Gestor de Bases de Datos Oracle esta vez en lenguaje SQL. Se consultará en la tabla SOL\_PRESTAMOS del museo al que pertenece el validador para devolver todos los pedidos que no hayan sido validados por el mismo.
- 4) El Gestor de Bases de Datos Oracle devuelve los datos pedidos en lenguaje SQL al componente Interface Base de Datos.
- 5) El componente “Interface Base de Datos” interpreta el lenguaje SQL y responde al Sistema Intermediario con el resultado que deseaba, para finalmente este mostrarlo por pantalla.



- 6) El Validador selecciona todas aquellas peticiones que desee validar con un OK y pulsa “Negar”
- 7) El Sistema Intermediario pide al componente denominado Interface Base de Datos que realice los cambios indicados por el validador en la BBDD.
- 8) El componente Interface Base de Datos a su vez realiza esa petición al Gestor de Bases de Datos Oracle esta vez en lenguaje SQL. Se insertará en la tabla SOL\_PRESTAMOS del museo al que pertenece el Validador para poner el campo validado a FF
- 9) El Gestor de Bases de Datos Oracle devuelve OK si todo ha ido correctamente al Interface Base de Datos.
- 10) El componente “Interface Base de Datos” interpreta el lenguaje SQL y responde al Sistema Intermediario con un ok si todo ha ido correctamente, y este muestra al Validador un mensaje en pantalla.

### 6.08 - Diagrama de Secuencia “DS008” Para el Caso de Uso “CU008”

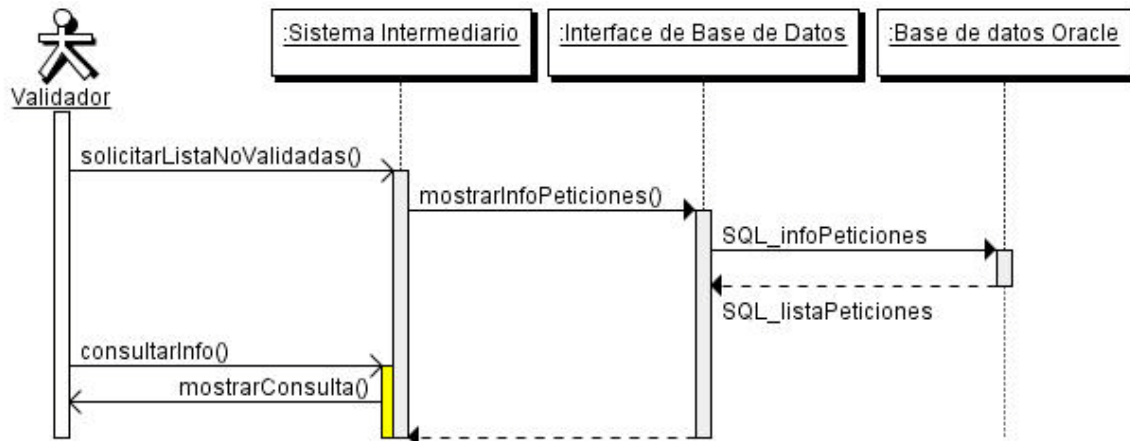


Figura 26: Diagrama de Secuencia “DS008”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU008” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “Validador” consulta la información de un determinado usuario.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El Validador selecciona en el menú lateral la opción “Validar Peticiones” en el Sistema Intermediario
- 2) El Sistema Intermediario pide al componente denominado Interface Base de Datos que obtenga un listado de todas las peticiones que han llegado al museo que aún no han sido validadas.
- 3) El componente Interface Base de Datos a su vez realiza esa petición al Gestor de Bases de Datos Oracle esta vez en lenguaje SQL. Se consultará en la tabla SOL\_PRESTAMOS del museo al que pertenece el validador para devolver todos los pedidos que no hayan sido validados por el mismo.
- 4) El Gestor de Bases de Datos Oracle devuelve los datos pedidos en lenguaje SQL al componente Interface Base de Datos.
- 5) El componente “Interface Base de Datos” interpreta el lenguaje SQL y responde al Sistema Intermediario con el resultado que deseaba, para finalmente este mostrarlo por pantalla.



- 6) El Validador pulsa sobre el NIF del usuario cuya información detallada desea conocer en profundidad.
- 7) El Sistema Intermediario le devuelve la información pedida

## 6.09 - Diagrama de Secuencia “DS009” Para el Caso de Uso “CU009”

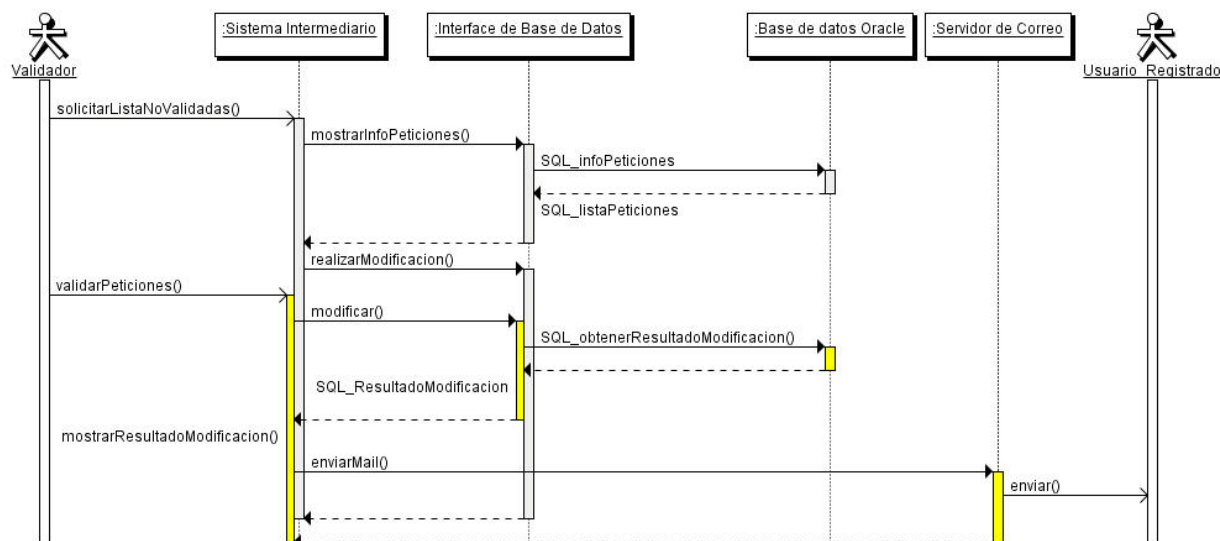


Figura 27: Diagrama de Secuencia “DS009”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU009” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando tras validar o negar el “Validador”, el sistema envía un mensaje por correo al “Usuario Registrado” que solicitó la petición

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El Validador selecciona en el menú lateral la opción “Validar Peticiones” en el Sistema Intermediario
- 2) El Sistema Intermediario pide al componente denominado Interface Base de Datos que obtenga un listado de todas las peticiones que han llegado al museo que aún no han sido validadas.
- 3) El componente Interface Base de Datos a su vez realiza esa petición al Gestor de Bases de Datos Oracle esta vez en lenguaje SQL. Se consultará en la tabla SOL\_PRESTAMOS del museo al que pertenece el validador para devolver todos los pedidos que no hayan sido validados por el mismo.
- 4) El Gestor de Bases de Datos Oracle devuelve los datos pedidos en lenguaje SQL al componente Interface Base de Datos.
- 5) El componente “Interface Base de Datos” interpreta el lenguaje SQL y responde al Sistema Intermediario con el resultado que deseaba, para finalmente este mostrarlo por pantalla.



- 6) El Validador selecciona todas aquellas peticiones que desee validar o negar y pulsa “Validar”/”Negar”
- 7) El Sistema Intermediario pide al componente denominado Interface Base de Datos que realice los cambios indicados por el validador en la BBDD.
- 8) El componente Interface Base de Datos a su vez realiza esa petición al Gestor de Bases de Datos Oracle esta vez en lenguaje SQL. Se insertará en la tabla SOL\_PRESTAMOS del museo al que pertenece el validador para poner el campo Validado a SI/FF
- 9) El Gestor de Bases de Datos Oracle devuelve OK si todo ha ido correctamente al Interface Base de Datos.
- 10) El componente “Interface Base de Datos” interpreta el lenguaje SQL y responde al Sistema Intermediario con un ok si todo ha ido correctamente, y este muestra al Validador un mensaje en pantalla.
- 11) El Sistema Intermediario envía al servidor de correo una petición para enviar un mail con la respuesta

### 6.10 - Diagrama de Secuencia “DS010” Para el Caso de Uso “CU010”

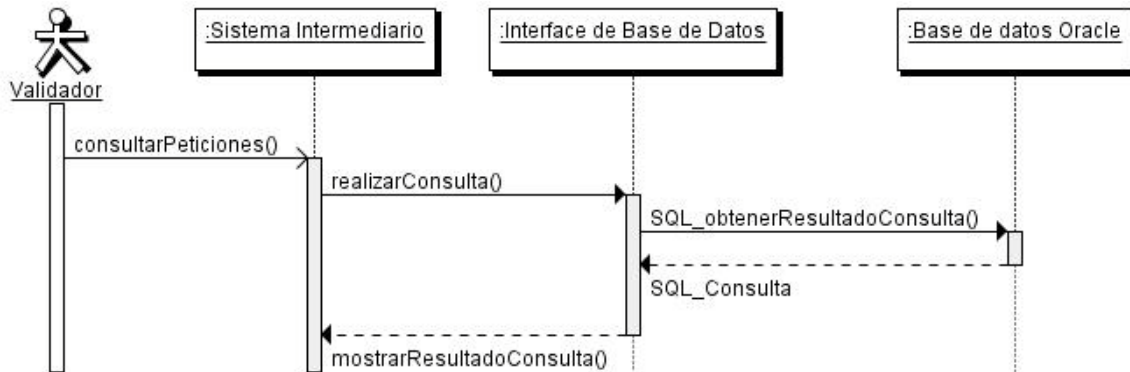


Figura 28: Diagrama de Secuencia “DS010”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU010” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “Validador” consulta todas las peticiones destinadas a su museo.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El Validador selecciona en el menú lateral la opción “Consultar Peticiones” en el Sistema Intermediario
- 2) El Sistema Intermediario pide al componente denominado Interface Base de Datos que obtenga la información que ha solicitado el usuario.
- 3) El componente Interface Base de Datos a su vez realiza esa petición al Gestor de Bases de Datos Oracle esta vez en lenguaje SQL. Se consultará en la tabla SOL\_PRESTAMOS todas las solicitudes del museo.
- 4) El Gestor de Bases de Datos Oracle devuelve los datos pedidos en lenguaje SQL al componente Interface Base de Datos.
- 5) El componente “Interface Base de Datos” interpreta el lenguaje SQL y responde al Sistema Intermediario con el resultado que deseaba, para finalmente este mostrarlo por pantalla.

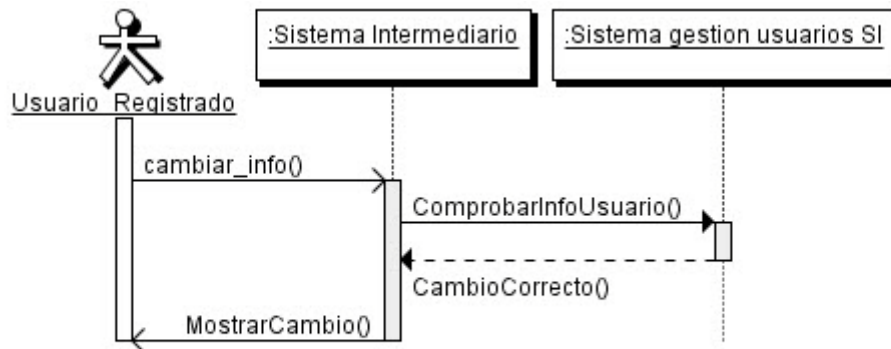
**6.11a - Diagrama de Secuencia “DS011a” Para el Caso de Uso “CU011”**

Figura 29: Diagrama de Secuencia “DS011a”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU011” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “Usuario\_Registrado” Cambia sus datos de contacto.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El Usuario Registrado selecciona en el menú derecho la opción “Cambiar Datos” en el Sistema Intermediario, posteriormente, introducirá los nuevos datos al sistema.
- 2) El Sistema Intermediario le enviará dicha información al Sistema de Gestión de Usuarios del S.I, el cual se encargará de comprobar la información introducida por el usuario.
- 3) El Sistema Gestor de Usuarios devuelve al Sistema Intermediario el resultado de la operación pedida
- 4) El Sistema Intermediario muestra dicha información por pantalla al Usuario



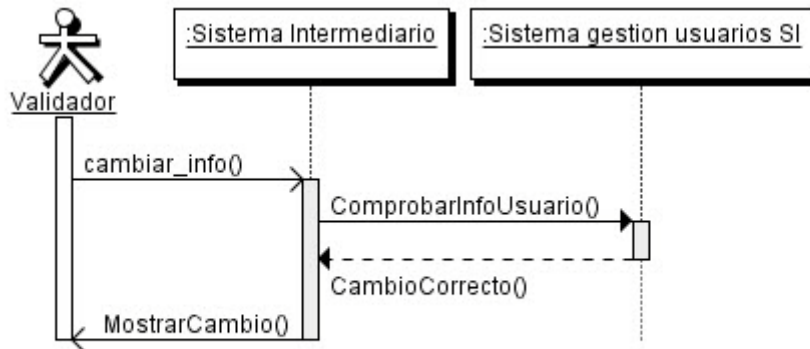
**6.11b - Diagrama de Secuencia “DS011b” Para el Caso de Uso “CU011”**

Figura 30: Diagrama de Secuencia “DS011b”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU011” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “Validador” Cambia sus datos de contacto.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 5) El Validador selecciona en el menú derecho la opción “Cambiar Datos” en el Sistema Intermediario, posteriormente, introducirá los nuevos datos al sistema.
- 6) El Sistema Intermediario le enviará dicha información al Sistema de Gestión de Usuarios del S.I, el cual se encargará de comprobar la información introducida por el usuario.
- 7) El Sistema Gestor de Usuarios devuelve al Sistema Intermediario el resultado de la operación pedida
- 8) El Sistema Intermediario muestra dicha información por pantalla al Validador

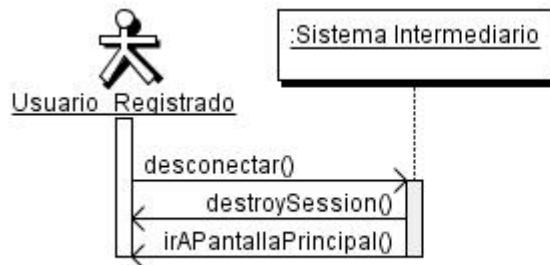
**6.12a - Diagrama de Secuencia “DS0012a” Para el Caso de Uso “CU0012”**

Figura 31: Diagrama de Secuencia “DS012a”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU012” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando el “Usuario\_Registrado” se desconecta del sistema.

Esta acción será una premisa para que el administrador pueda realizar cualquier otra acción con privilegios dentro del “Back-End”.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 1) El Usuario\_Registrado pulsa el botón desconectar de su menú lateral izquierdo en el Sistema Intermediario
- 2) El Sistema Intermediario destruye su sesión y vuelve a establecerle como usuario anónimo
- 3) El Sistema Intermediario vuelve a la página principal de presentación de la aplicación. El usuario podrá logarse nuevamente si así lo desea.

### 6.12b - Diagrama de Secuencia “DS0012b” Para el Caso de Uso “CU0012”

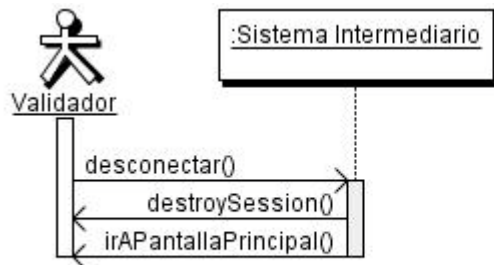


Figura 32: Diagrama de Secuencia “DS012b”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU012” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando el “Validador” se desconecta del sistema.

Esta acción será una premisa para que el administrador pueda realizar cualquier otra acción con privilegios dentro del “Back-End”.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

- 4) El Validador pulsa el botón desconectar de su menú lateral izquierdo en el Sistema Intermediario
- 5) El Sistema Intermediario destruye su sesión y vuelve a establecerle como usuario anónimo
- 6) El Sistema Intermediario vuelve a la página principal de presentación de la aplicación. El usuario podrá logarse nuevamente si así lo desea.



## 7 - TRAZABILIDAD

Trazabilidad significa capturar las relaciones de implementación y dependencia en el modelo, siguiendo un histórico de los pasos seguidos para desarrollar el software.

Por ejemplo:

- 1) Un proceso de negocio requerirá estudiarlo para poder sacar del cliente que nos reclama dicho software una serie de “*REQUISITOS*”.
- 2) A partir de dichos requisitos definiremos las principales funcionalidades del sistema “*CASOS DE USO*” para implementar las funciones del proceso.
- 3) De estudiar, dividiendo en pasos y tareas estos casos de uso, surgirán más tarde los “*DIAGRAMAS DE SECUENCIA*”.

El proceso anterior nos hace ver una serie de pasos que van avanzando desde un lenguaje cercano a nuestro (lenguaje natural) hacia un lenguaje más cercano al software.

En el caso del software desarrollado en este proyecto, un ejemplo de la trazabilidad desde un requisito hasta su diagrama de secuencia correspondiente sería el siguiente:

REQUISITO ( <b>RUC001</b> ) → CASO DE USO ( <b>CU001</b> ) → DIAGRAMA DE SECUENCIA ( <b>DS001</b> )
---



## 7.1 – Tabla de Trazabilidad

TABLA DE TRAZABILIDAD		
REQUISITOS DE CAPACIDAD	CASOS DE USO	DIAGRAMAS DE SECUENCIA
RUC001	CU001	DS001
RUC002	CU002	DS002
RUC003	CU003	DS003a
RUC003	CU003	DS003b
RUC004	CU004	DS004
RUC005	CU005	DS005
RUC006	CU006	DS006
RUC007	CU007	DS007a
RUC007	CU007	DS007b
RUC008	CU008	DS008
RUC009	CU009	DS009
RUC0010	CU0010	DS0010
RUC011	CU011	DS011a
RUC011	CU011	DS011b
RUC012	CU012	DS012a
RUC012	CU012	DS012b

Figura 33: Tabla de Trazabilidad

## 8 - DISEÑO GRÁFICO DE LA APLICACIÓN WEB

Vamos a hacer una descripción gráfica del diseño de la aplicación web, de modo que se vaya viendo, paso a paso, cómo se ha realizado la misma

### 8.01 - DISEÑO DE LA PÁGINA PRINCIPAL



Figura 34: Diseño gráfico de la portada

Como podemos observar en la captura de la portada, vemos que la distribución del portal esta dividida por zonas, siguiendo este esquema:

- En la parte superior: El Título principal de la página. Es totalmente configurable y debería aparecer el nombre de la aplicación. Se ha decidido darle un fondo gris con letras blancas para resaltarlo.
- En la parte central: Será donde vayan apareciendo todos los contenidos de nuestra aplicación, no obstante, para la portada, se ha elegido una reproducción del cuadro de Rembrandt “Noche de Ronda” puesto que encaja perfectamente con la temática de la aplicación y por su belleza estética general.
- En el lateral izquierdo: Antes de la validación por parte del usuario, el menú lateral izquierdo aparecerá desactivado. Al conectarse el usuario al Sistema, dicho menú se activará, pudiendo el usuario acceder a sus opciones.



- En el lateral derecho: Será donde aparezca el cuadro de Login al sistema. Cuando el usuario se haya logado, dicho cuadro se sustituirá por un mensaje de bienvenida donde se le informará de su mail y teléfono actual.
- En la parte inferior: La parte inferior es la zona de logos. Se introducirán en dicha parte todos los logos que sean necesarios, contador de visitas, información de contacto....

## 8.02 - DISEÑO DE LA PÁGINA PRINCIPAL DEL USUARIO REGISTRADO

The screenshot shows a web application titled "Rembrandt Inter-Museos". The interface is divided into three main sections:

- Left Sidebar:** Contains a vertical menu with the following options: "Consultar Obras", "Modificar Pedido", "Solicitar Pedido", "Ver Pedidos Realizados", and a "Desconectar" button at the bottom.
- Central Area:** Features the heading "Introduzca aquí su consulta". Below this is a search input field containing the text "mar" and a "Consultar" button.
- Right Sidebar:** Displays a welcome message "Bienvenido al Sistema" followed by the text "Está usted conectado/a con el usuario admin". It also shows the user's contact information: "Mail Actual: pepe@marduki.com" and "Teléfono Actual: 988777666". A "Cambiar Datos" button is located at the bottom of this section.

At the bottom of the page, there is a link for "Información de Contacto". The browser's address bar shows the URL "http://localhost:48232/Proyecto\_FC/ServletControlador?opc=a", and the status bar indicates "Internet | Modo protegido: activado" and a zoom level of "100%".

Figura 35: Página Principal de Usuario Registrado

Para todo el portal, se ha utilizado como color de fondo el color Plata, para dar mayor aspecto corporativo al mismo. Asimismo se producen los siguientes cambios en el sistema:

- Como se puede observar, la parte izquierda ha sido sustituida por el menú desplegable con las opciones para un usuario registrado del sistema.
- La parte derecha ha sido sustituida por los datos del usuario junto con la opción Cambiar Datos.
- Como hemos mencionado anteriormente, toda la información del usuario aparecerá en la pantalla central, siendo los menús laterales opcionales para mejorar la navegabilidad.

### 8.03 - DISEÑO DE LA PÁGINA PRINCIPAL DEL VALIDADOR



The screenshot shows the 'Rembrandt Inter-Museos' interface for a validator. It features a left sidebar with buttons for 'Validar Pedidos', 'Mostrar Pedidos', and 'Desconectar'. The main area is titled 'Resultado de su consulta' and contains a table with columns: 'Seleccionar', 'Obra', 'Pedido', 'Usuario', 'Fecha Inicio', and 'Fecha Fin'. A single row is displayed with values: a checkbox, 'AA33332', '40', 'valid@marduki.es', '14/4/2010', and '11/2/2015'. Below the table are 'Validar' and 'Negar' buttons. The right sidebar shows connection status, email, phone, and a 'Cambiar Datos' button. A footer link 'Información de Contacto' is present.

Seleccionar	Obra	Pedido	Usuario	Fecha Inicio	Fecha Fin
<input type="checkbox"/>	AA33332	40	valid@marduki.es	14/4/2010	11/2/2015

Figura 36: Página principal del Validador

- Tal como sucedía con el usuario registrado, la parte izquierda ha sido sustituida por el menú desplegable con las opciones para validador del sistema.
- La parte derecha ha sido sustituida por los datos del validador junto con la opción Cambiar Datos.
- Tal como sucedía con el usuario registrado, toda la información del validador aparecerá en la pantalla central, siendo los menús laterales opcionales para mejorar la navegabilidad.

Cabe indicar además con respecto al estilo que se ha realizado una plantilla CSS (plantilla de apariencia) de modo que todo sea visible fácilmente, con la finalidad de que se diferencien bien las secciones y que el usuario no se vea desbordado por un sinnúmero de colores y fuentes.

Asimismo, se ha hecho especial énfasis en el diseño de los menús laterales, puesto que uno de los principales objetivos del portal es que esos menús y opciones laterales estén siempre a la vista, de modo que al navegar sólo cambie el contenido central (la información en cuestión). Para la realización de los mismos, se ha utilizado el programa CSS TAB DESIGNER



## 8.04 - DISEÑO DEL MENÚ PRINCIPAL

El menú principal, como ya dijimos, es la principal herramienta de navegación por el portal. Todos los elementos del menú permiten una navegación mas cómoda y fácil, así como acceso inmediato a las principales opciones de la aplicación.

Inicialmente, las opciones del menú aparecerán desactivadas



Figura 37: Menú Usuario Desactivado

No obstante, tras realizar log-in en el sistema como usuario o validador, las opciones desactivadas pasarán a activarse dependiendo del tipo de usuario que se haya conectado



Figura 38: Menú Usuario



Figura 39: Menú Validador

En el diseño gráfico de los menús (plantilla CSS de estilos realizada para este proyecto) se ha elegido un fondo negro con letras blancas para resaltar la información.

Asimismo, debajo del menú iría la opción desconectar, por si el usuario desea cerrar sesión en el sistema. Desconectar es un botón, y por lo tanto se le aplicará el formato estándar de button J2EE.



- Las opciones del menú tendrán un fondo negro como ya se comentó anteriormente para poder resaltar la información al usuario de la aplicación, como se muestra en la primera representación gráfica de la “Figura 40”.



- Al pasar el puntero del ratón sobre las opciones del menú, estas resaltarán sobre un fondo, negro más oscuro, como se muestra en la segunda representación gráfica de la “Figura 40”.

## 8.05 - DISEÑO DEL FORMULARIO DE ACCESO

El formulario de acceso de usuarios nos permite autenticarnos ante el sistema por medio de nuestro Usuario y Clave, como muestra la “Figura 43”.



Por favor autentíquese

Formulario de acceso con campos para Usuario y Clave\*\*, y un botón Login.

Figura 40: Formulario de Acceso

Cabe destacar que la contraseña aparecerá con \*\*\* a fin de ocultar la misma a posibles personas malintencionadas.

## 8.06 - DISEÑO DEL MENÚ DE BIENVENIDA

En el menú lateral derecho aparece el formulario de acceso en la pantalla principal cuando el usuario aún no se ha registrado en el sistema.



Por favor autentíquese

Formulario de acceso con campos para Usuario y Clave\*\*, y un botón Login.

Figura 41: Formulario de Acceso



Cuando el usuario esté autenticado, dicho formulario se cambiará por el siguiente mensaje:



Figura 42: Mensaje lateral derecho

Donde se indicará al usuario el teléfono y mail actual con los que está actualmente en el sistema, y se le permitirá cambiarlos en caso de que lo desee.

## 8.07 - DISEÑO DE MENÚ INFERIOR

El menú inferior está pensado para Logos, contador de visitas etc.... Se ha implementado con un logo muy básico, pero se puede ajustar dependiendo de las entidades que participen en el sistema.

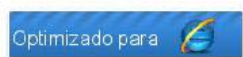


Figura 43: Logos

El espacio para los mismos se ha dejado deliberadamente amplio para poder introducir todos los logos que sean necesarios. En la figura de la parte superior únicamente se muestra un logo de ejemplo para dar una idea general del aspecto de la aplicación.



## 9 - BATERÍA DE PRUEBAS REALIZADAS AL SISTEMA

Para realizar la descripción de las distintas pruebas que se llevarán a cabo durante el proceso de desarrollo del sistema, se ha definido un modelo de tabla en el que se desarrollarán las siguientes características de cada una de las pruebas:

- **Indicador:** Nomenclatura con la que la prueba a describir será identificada, esta contará de las iniciales P y de una numeración que será unívoca para cada una.
- **Objetivo:** Se indicará la utilidad de la prueba propuesta y el fin por el que se debe realizar.
- **Subsistema:** Se indicará el subsistema al que estará relacionada la prueba, este será modelo, vista o controlador.
- **Método:** Se indicará el método, dentro de cada componente, al que estará relacionada la prueba descrita.
- **Secuencia de realización:** Se indicarán los pasos a seguir de forma ordenada, para realizar la prueba a comprobar en cada caso.
- **Entrada:** Se indicará la entrada al método necesaria para que la prueba se realice con éxito.
- **Salida válida:** Se indicará la salida necesaria para comprobar que la prueba se ha realizado con éxito.

### 9.1 Pruebas de Usuario

Identificador	P-001
<b>Objetivo</b>	Comprobar el usuario/clave que el usuario introduce cuando desea hacer log-in en el sistema
<b>Subsistema</b>	Vista
<b>JSP</b>	Index.jsp, enlaces.jsp
<b>Secuencia de realización</b>	<ol style="list-style-type: none"> <li>1. Usuario introduce su DNI y su clave en el sistema</li> <li>2. Usuario pulsa Login</li> <li>3. Si el login es correcto -&gt; Acceso al sistema</li> <li>4. Else -&gt; Se muestra mensaje de error a usuario</li> </ol>
<b>Entrada</b>	Usuario/clave
<b>Salida válida</b>	TRUE en caso de ser usuario/clave válidos FALSE en caso contrario



Identificador	P-002
Objetivo	Consultar las obras de arte en todas las BBDD de todos los museos incluidos en nuestro sistema a partir de un determinado patrón
Subsistema	Vista
JSP	Consultas.jsp
Secuencia de realización	<div><div><div>1. Usuario introduce el fragmento que desea consultar</div><div>2. Usuario pulsa Consultar</div><div>3. Si hay obras que mostrar -&gt; se muestran al usuario los siguientes campos para todas las obras:</div><div><div>Seleccionar</div><div>ID Obra</div><div>Fecha</div><div>Título</div><div>Autor</div><div>Prestado</div></div><div>4. Else -&gt; Se muestra mensaje a usuario “No hay ningun resultado para su búsqueda”</div></div></div>
Entrada	Introducir clave en caja de texto y pulsar CONSULTAR
Salida válida	TRUE en caso de ser usuario/contraseña válidos FALSE en caso contrario

Identificador	P-003				
Objetivo	Consultar información adicional de una obra de arte				
Subsistema	Vista				
JSP	MostrarRestoInfoObra.jsp				
Secuencia de realización	<div>1. Partiendo del caso anterior, usuario pulsa sobre el id de la obra de arte cuya información adicional desea conocer</div> <div>2. Deberán aparecer listados los siguientes campos:</div>				
	ID Obra	Nombre Prestador	Técnica	Medidas	Dirección de Recogida
	Dirección de Devolución	Valor del seguro	Observaciones del seguro	Correo	
Entrada	Click sobre el ID_Obra que se desea ver				
Salida válida	Siempre True				



Identificador	P-004
Objetivo	Volver a menú anterior desde la consulta adicional de una obra de arte
Subsistema	Vista
JSP	MostrarRestoInfoObra.jsp
Secuencia de realización	<ol style="list-style-type: none"><li>1. Partiendo del caso anterior, usuario deberá pulsar el botón VOLVER</li><li>2. Usuario volverá a los resultados de su consulta</li></ol>
Entrada	Pulsar VOLVER
Salida válida	Siempre True

Identificador	P-005
Objetivo	Volver a menú anterior desde la pantalla de resultados de la consulta de una obra de arte
Subsistema	Vista
JSP	ImprimirLista.jsp
Secuencia de realización	<ol style="list-style-type: none"><li>1. En la pantalla de resultado de la consulta el usuario pulsa el botón VOLVER</li><li>2. Usuario volverá a la pantalla de consulta de una obra de arte</li></ol>
Entrada	Pulsar VOLVER
Salida válida	Siempre True





Identificador	P-006
<b>Objetivo</b>	Seleccionar una o varias obras y pulsar Añadir a Pedido para meter dichas obras en el pedido
<b>Subsistema</b>	Vista
<b>JSP</b>	ImprimirLista.jsp
<b>Secuencia de realización</b>	<ol style="list-style-type: none"> <li>1. En la pantalla de resultado de la consulta el usuario selecciona una/varias obras de arte.</li> <li>2. Usuario pulsa Añadir a Pedido</li> <li>3. Usuario accederá a la pantalla de su pedido con todas las obras de arte seleccionadas hasta el momento</li> </ol>
<b>Entrada</b>	Seleccionar una o varias obras y pulsar Añadir a Pedido
<b>Salida válida</b>	Siempre True

Identificador	P-007
<b>Objetivo</b>	Seleccionar una o varias obras y pulsar Añadir a Pedido para meter dichas obras en el pedido
<b>Subsistema</b>	Vista
<b>JSP</b>	ImprimirLista.jsp
<b>Secuencia de realización</b>	<ol style="list-style-type: none"> <li>1. En la pantalla de resultado de la consulta el usuario selecciona una/varias obras de arte.</li> <li>2. Usuario pulsa Añadir a Pedido</li> <li>3. Usuario accederá a la pantalla de su pedido con todas las obras de arte seleccionadas hasta el momento</li> </ol>
<b>Entrada</b>	Seleccionar una o varias obras y pulsar Añadir a Pedido
<b>Salida válida</b>	Siempre True



Identificador	P-008
<b>Objetivo</b>	Eliminar obras del pedido
<b>Subsistema</b>	Vista
<b>JSP</b>	ImprimirCarro.jsp
<b>Secuencia de realización</b>	<ol style="list-style-type: none"> <li>1. En la pantalla del pedido, el usuario selecciona una/varias obras de arte.</li> <li>2. Usuario pulsa Eliminar</li> <li>3. Sistema mostrará al usuario su nueva selección de obras, que consistirá en las que tenía anteriormente, menos las que ha eliminado</li> </ol>
<b>Entrada</b>	Seleccionar una o varias obras y pulsar Eliminar
<b>Salida válida</b>	Siempre True

Identificador	P-009
<b>Objetivo</b>	Consultar más Obras
<b>Subsistema</b>	Vista
<b>JSP</b>	ImprimirCarro.jsp
<b>Secuencia de realización</b>	<ol style="list-style-type: none"> <li>1. En la pantalla del pedido, el usuario pulsa el botón Consultar más Obras</li> <li>2. Sistema llevará al usuario de nuevo a la pantalla de consulta de obras de arte</li> </ol>
<b>Entrada</b>	Pulsar el botón Consultar más Obras
<b>Salida válida</b>	Siempre True



Identificador	P-010
<b>Objetivo</b>	Solicitar Obras
<b>Subsistema</b>	Vista
<b>JSP</b>	ImprimirCarro.jsp
<b>Secuencia de realización</b>	<ol style="list-style-type: none"> <li>1. En la pantalla de pedidos, el usuario pulsa el botón Solicitar</li> <li>2. Sistema llevará al usuario ante la pantalla de establecimiento de fechas de préstamo</li> </ol>
<b>Entrada</b>	Pulsar el botón Solicitar en la pantalla de pedidos
<b>Salida válida</b>	Siempre True

Identificador	P-011
<b>Objetivo</b>	Volver a la pantalla de pedido
<b>Subsistema</b>	Vista
<b>JSP</b>	establecerfechasdevol.jsp
<b>Secuencia de realización</b>	<ol style="list-style-type: none"> <li>1. En la pantalla de establecimiento de fechas, el usuario pulsa el botón Volver</li> <li>2. Sistema llevará al usuario de nuevo ante la pantalla de pedidos de Obras de Arte</li> </ol>
<b>Entrada</b>	Pulsar el botón Volver en la pantalla de establecimiento de fechas
<b>Salida válida</b>	Siempre True



Identificador	P-012
<b>Objetivo</b>	Establecer las fechas del sistema
<b>Subsistema</b>	Vista
<b>JSP</b>	establecerfechasdevol.jsp
<b>Secuencia de realización</b>	<ol style="list-style-type: none"> <li>1. En la pantalla de establecimiento de fechas, para cada obra, el usuario Selecciona la fecha en la que desee que comience dicho préstamo. Dicha fecha deberá ser al menos diez días posteriores al actual para poder realizar los preparativos.</li> <li>2. Usuario selecciona la fecha en la que desea que finalice dicho préstamo</li> <li>3. Usuario marca con un tick las obras de arte que incluirá en la solicitud final</li> <li>4. Usuario pulsa Solicitar</li> <li>5. Si todo es correcto -&gt; El pedido se insertará en las diferentes BBDD de los museos y se mostrará al usuario mensaje de éxito.</li> <li>6. En caso contrario se mostrará mensaje de error</li> </ol>
<b>Entrada</b>	Pulsar el botón Solicitar en la pantalla de establecimiento de fechas tras haber seguido los pasos anteriormente indicados
<b>Salida válida</b>	Si todo es correcto True. Else False

Identificador	P-013
<b>Objetivo</b>	Cambiar los Datos Personales del Usuario
<b>Subsistema</b>	Vista
<b>JSP</b>	cambiardatos.jsp
<b>Secuencia de realización</b>	<ol style="list-style-type: none"> <li>1. En la pantalla derecha usuario pulsará sobre la opción Cambiar Datos en la pantalla derecha</li> <li>2. Sistema Solicitará a usuario cambiar teléfono y mail</li> <li>3. Si usuario pulsa cancelar volverá a la pantalla derecha anterior</li> <li>4. Usuario deberá Introducir nuevo mail y teléfono</li> <li>5. Si todo es correcto -&gt; Se realizará el cambio y se le mostrará un mensaje por pantalla al usuario</li> <li>6. En caso contrario-&gt;Se le indicará el motivo del error al usuario</li> </ol>
<b>Entrada</b>	Pulsar el botón Cambiar Datos en la pantalla derecha
<b>Salida válida</b>	Todo correcto True. Else false. Si usuario pulsa Volver volverá a pantalla anterior.



Identificador	P-014												
Objetivo	Ver los pedidos realizados por el usuario												
Subsistema	Vista												
JSP	imprimirconsultapeticiones.jsp												
Secuencia de realización	1. Usuario selecciona Ver pedidos Realizados en el menú izquierdo												
	2. Si tiene alguno -> Sistema muestra información sobre el mismo. La información mostrada será												
	<table><tr><td>Obra</td><td>Pedido</td><td>Titulo</td><td>Usuario</td><td>Inicio de Prestamo</td><td>Fin de Prestamo</td><td>Validado</td></tr></table>							Obra	Pedido	Titulo	Usuario	Inicio de Prestamo	Fin de Prestamo
Obra	Pedido	Titulo	Usuario	Inicio de Prestamo	Fin de Prestamo	Validado							
	3. Si no tiene ninguno mostrará mensaje “No hay ningún resultado para su consulta”												
Entrada	Pulsar el botón Ver Pedidos Realizados en la pantalla izquierda												
Salida válida	True o mensaje de error												

Identificador	P-015
<b>Objetivo</b>	Desconexión
<b>Subsistema</b>	Vista
<b>JSP</b>	enlaces.jsp
<b>Secuencia de realización</b>	<ol style="list-style-type: none"> <li>1. Usuario pulsa Desconectar en el menú izquierdo</li> <li>2. Su sesión es terminada y vuelve a la pantalla principal</li> </ol>
<b>Entrada</b>	Pulsar el botón Desconectar en la pantalla izquierda
<b>Salida válida</b>	Siempre True



Identificador	P-016
Objetivo	Cese de la sesión
Subsistema	Controlador
JSP	-
Secuencia de realización	1. Si el usuario está conectado 10 o más minutos sin hacer nada, al intentar realizar cualquier función del sistema, este le indicará que su sesión ha caducado
Entrada	10 minutos de inactividad
Salida válida	Siempre FALSE



## 9.2 Pruebas de Validador

Identificador	P-017
Objetivo	Mostrar Todos los pedidos validados o pendientes de validar por el validador
Subsistema	Vista
JSP	imprimirtodaslassolicitudes.jsp
Secuencia de realización	<div><div>1. Validador selecciona Mostrar pedidos en el menú izquierdo</div><div>2. Al Validador se le muestran los siguientes datos:</div><div><div>Obra</div><div>Pedido</div><div>Usuario</div><div>Fecha Inicio</div><div>Fecha Fin</div><div>Validado</div></div></div>
Entrada	Seleccionar Mostrar pedidos en el menú izquierdo
Salida válida	Siempre True

Identificador	P-018
Objetivo	Ver mas información sobre el usuario que realizó la petición
Subsistema	Vista
JSP	Mostrarrestoinfousuario.jsp
Secuencia de realización	<div><div><div>1. Validador pulsa sobre el DNI del usuario cuya información adicional desea consultar</div><div>2. Al validador se le muestran los siguientes datos:</div><div><div>NIF Usuario</div><div>Nombre</div><div>Mail</div><div>Direccion</div><div>Telefono</div></div><div>3. Para volver a la pantalla anterior, deberá pulsar Volver</div></div></div>
Entrada	Pulsar sobre el DNI del usuario cuya información adicional desea consultar
Salida válida	Siempre True



Identificador	P-019
<b>Objetivo</b>	Validar un pedido
<b>Subsistema</b>	Vista
<b>JSP</b>	Imprimirnovalidadas.jsp
<b>Secuencia de realización</b>	<ol style="list-style-type: none"> <li>Validador pulsa sobre la opción Validar Pedidos del menú lateral izquierdo</li> <li>Si tiene alguna solicitud pendiente de validar -&gt; Se le mostrará la siguiente información:</li> <li>Si no -&gt; Se le mostrará el Mensaje (“No hay ningún resultado para su consulta”)</li> <li>En caso de tener una o varias solicitudes pendientes que desee validar, deberá seleccionar todas las que desee validar y pulsar sobre el botón Validar</li> <li>Usuario recibirá mensaje de confirmación</li> </ol>
<b>Entrada</b>	Pulsa sobre la opción Validar Pedidos del menú lateral izquierdo, seleccionar solicitudes a validar y Pulsar validar
<b>Salida válida</b>	Siempre True excepto que no haya solicitudes pendientes

Identificador	P-020
<b>Objetivo</b>	Negar un pedido
<b>Subsistema</b>	Vista
<b>JSP</b>	Imprimirnovalidadas.jsp
<b>Secuencia de realización</b>	<ol style="list-style-type: none"> <li>Validador pulsa sobre la opción Validar Pedidos del menú lateral izquierdo</li> <li>Si tiene alguna solicitud pendiente de validar -&gt; Se le mostrará la siguiente información:</li> <li>Si no -&gt; Se le mostrará el Mensaje (“No hay ningún resultado para su consulta”)</li> <li>En caso de tener una o varias solicitudes pendientes que desee negar, deberá seleccionar todas las que desee validar y pulsar sobre el botón Negar</li> <li>Usuario recibirá mensaje de confirmación</li> </ol>
<b>Entrada</b>	Pulsa sobre la opción Validar Pedidos del menú lateral izquierdo, seleccionar solicitudes a validar y Pulsar negar
<b>Salida válida</b>	Siempre True excepto que no haya solicitudes pendientes





Identificador	P-021
<b>Objetivo</b>	Envío de mensaje de validación/negación a usuario
<b>Subsistema</b>	Controlador
<b>JSP</b>	-
<b>Secuencia de realización</b>	Al validar o negar, automáticamente el usuario que ha realizado la solicitud deberá recibir un mail a su cuenta de correo del sistema indicándosele que se le ha negado o validado el pedido.
<b>Entrada</b>	Aceptación o negación de la solicitud
<b>Salida válida</b>	Si existe la cuenta de correo en el servidor de correo y el mensaje se puede enviar correctamente -> True en caso contrario False

Identificador	P-022
<b>Objetivo</b>	Cese de la sesión
<b>Subsistema</b>	Controlador
<b>JSP</b>	-
<b>Secuencia de realización</b>	1. Si el validador está conectado 10 o más minutos sin hacer nada, al intentar realizar cualquier función del sistema, este le indicará que su sesión ha caducado
<b>Entrada</b>	10 minutos de inactividad
<b>Salida válida</b>	Siempre FALSE



Identificador	P-023
<b>Objetivo</b>	Desconexión
<b>Subsistema</b>	Vista
<b>JSP</b>	enlaces.jsp
<b>Secuencia de realización</b>	<ol style="list-style-type: none"> <li>Validador pulsa Desconectar en el menú izquierdo</li> <li>Su sesión es terminada y vuelve a la pantalla principal</li> </ol>
<b>Entrada</b>	Pulsar el botón Desconectar en la pantalla izquierda
<b>Salida válida</b>	Siempre True

Identificador	P-024
<b>Objetivo</b>	Cambiar Datos Personales
<b>Subsistema</b>	Vista
<b>JSP</b>	cambiardatos.jsp
<b>Secuencia de realización</b>	<ol style="list-style-type: none"> <li>En la pantalla derecha Validador pulsará sobre la opción Cambiar Datos en la pantalla derecha</li> <li>Sistema Solicitará a Validador cambiar teléfono y mail</li> <li>Si Validador pulsa cancelar volverá a la pantalla derecha anterior</li> <li>Validador deberá Introducir nuevo mail y teléfono</li> <li>Si todo es correcto -&gt; Se realizará el cambio y se le mostrará un mensaje por pantalla al usuario</li> <li>En caso contrario-&gt;Se le indicará el motivo del error al usuario</li> </ol>
<b>Entrada</b>	Pulsar el botón Cambiar Datos en la pantalla derecha
<b>Salida válida</b>	Todo correcto True. Else false. Si Validador pulsa Volver volverá a pantalla anterior.



### 9.3 Estado Final de las pruebas

A continuación indicamos el estado final de cada una de las pruebas indicadas anteriormente

Identificador	Objetivo	Resultado de la Prueba
P-001	Comprobar el usuario/clave que el usuario introduce cuando desea hacer log-in en el sistema	OK
P-002	Consultar las obras de arte en todas las BBDD de todos los museos incluidos en nuestro sistema a partir de un determinado patrón	OK
P-003	Consultar información adicional de una obra de arte	OK
P-004	Volver a menú anterior desde la consulta adicional de una obra de arte	OK
P-005	Volver a menú anterior desde la pantalla de resultados de la consulta de una obra de arte	OK
P-006	Seleccionar una o varias obras y pulsar Añadir a Pedido para meter dichas obras en el pedido	OK
P-007	Seleccionar una o varias obras y pulsar Añadir a Pedido para meter dichas obras en el pedido	OK
P-008	Eliminar obras del pedido	OK
P-009	Consultar más Obras	OK
P-010	Solicitar Obras	OK
P-011	Volver a la pantalla de pedido	OK
P-012	Establecer las fechas del sistema	OK
P-013	Cambiar los Datos Personales del Usuario	OK
P-014	Ver los pedidos realizados por el usuario	OK
P-015	Desconexión	OK
P-016	Cese de la sesión	OK
P-017	Mostrar Todos los pedidos validados o pendientes de validar por el validador	OK
P-018	Ver mas información sobre el usuario que realizó la petición	OK
P-019	Validar un pedido	OK
P-020	Negar un pedido	OK
P-021	Envío de mensaje de validación/negación a usuario	OK
P-022	Cese de la sesión	OK
P-023	Desconexión	OK
P-024	Cambiar Datos Personales	OK





## 10 – CONCLUSIONES Y TRABAJOS FUTUROS

### 10.1 Conclusiones

Como hemos visto anteriormente, se han cumplido los principales objetivos que nos habíamos propuesto.

- Se ha creado una aplicación JAVA robusta y de fácil manejo, cuya principal función es la de facilitar el préstamo de obras de arte entre los diversos museos.
- Además, dicha herramienta servirá no sólo para la solicitud de los préstamos, sino también para la validación de los mismos y para el intercambio de información entre prestador y usuario solicitante.
- La aplicación está dotada de un mecanismo de seguridad de tipo usuario/contraseña que impedirá a cualquier persona no autorizada hacer uso fraudulento de la misma.
- Asimismo, en caso de ser configurada, la aplicación podrá hacer uso de un servidor de correos para enviar al usuario un mail cada vez que sea aceptado o negado uno de sus préstamos por parte del validador del museo que está actualmente en posesión de la obra que se va a solicitar.
- Se ha incluido además un manual de usuario muy gráfico para facilitar el manejo de la misma a sus futuros usuarios.

Por último, indicar que dado que se han utilizado patrones de diseño a la hora de su implementación, es una aplicación abierta y fácilmente modificable para cualquier mejora posterior.

### 10.2 Trabajos Futuros

A continuación indicamos posibles mejoras que se podrían realizar a nuestro sistema, siendo respetuosos con el diseño original, pero mejorando diversos aspectos del mismo:

- Sistema de backups para los ficheros: Se debería crear un sistema de backups general para todos los ficheros del sistema. De esta manera, se minimizaría el impacto de cualquier posible incidencia en el mismo.
- Utilización de un sistema de identificación mediante certificados digitales: Actualmente nuestra aplicación utiliza un sencillo sistema usuario/contraseña. Aconsejaríamos para mejorar la seguridad de la misma en un futuro utilizar un sistema de validación mediante certificados digitales además del clásico usuario/contraseña.
- Base de datos para el Sistema Intermediario: Si bien sería más lento, podría ser muy interesante utilizar una BBDD para gestionar el Sistema Intermediario, puesto que se mejoraría el módulo de usuarios y museos, y se podrían implementar funcionalidades diversas
- Imágenes de las obras de arte: Como consecuencia de lo anterior, se podría incluso meter imágenes o fotos de las diferentes obras de arte, ya que podríamos utilizar la BBDD para gestionar en que directorio se encuentra cada imagen.



## 11 - BIBLIOGRAFÍA

Estándar de Ingeniería del Software “ESA PSS-05-0”:

<http://www.esa.int/esaCP/index.html>

Métrica V3:

<http://www.csae.map.es/csi/metrica3/index.html>

Patrón modelo vista-controlador:

<http://www.proactiva-calidad.com/java/patrones/mvc.html>

JSTL

<http://java.sun.com/products/jsp/jstl/>

Oracle 10G

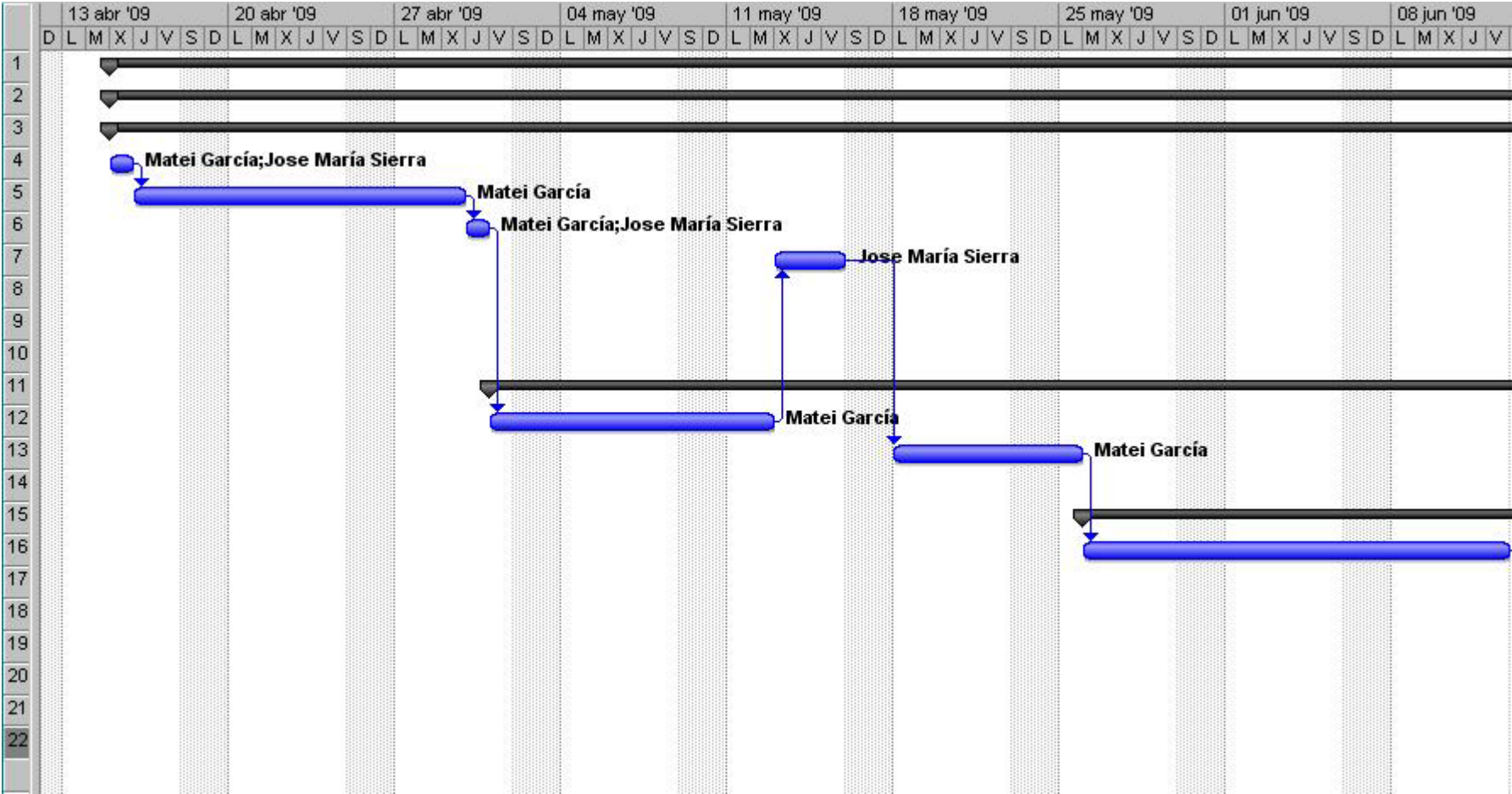
Perez, César., Oracle 10g. Administración y Análisis de Bases de Datos. 2ª edición , Ra-MA, 2007

# ANEXO 1 – PRESUPUESTO Y PLANIFICACIÓN

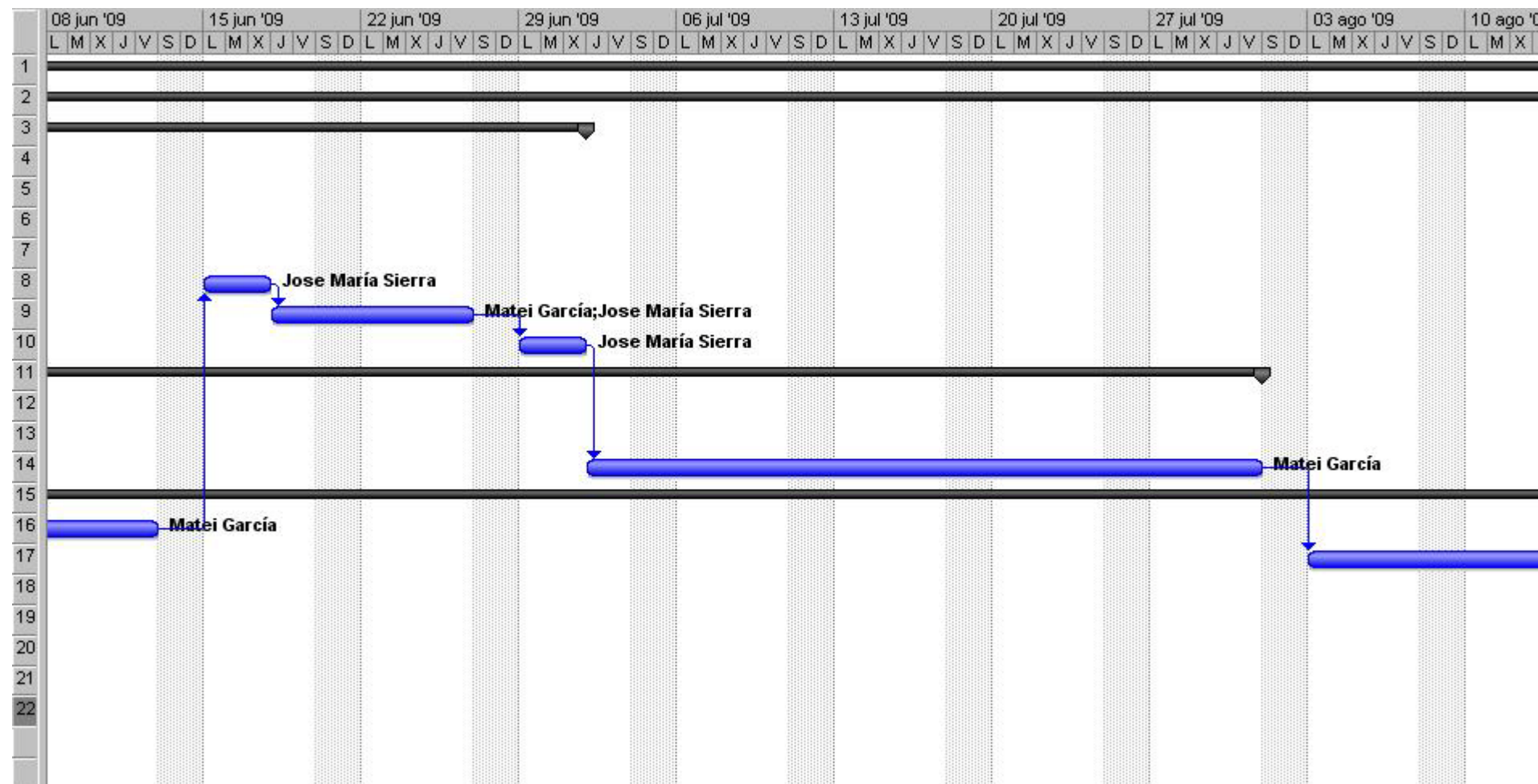
## Anexo 1.1 Gantt del Proyecto Diagramas

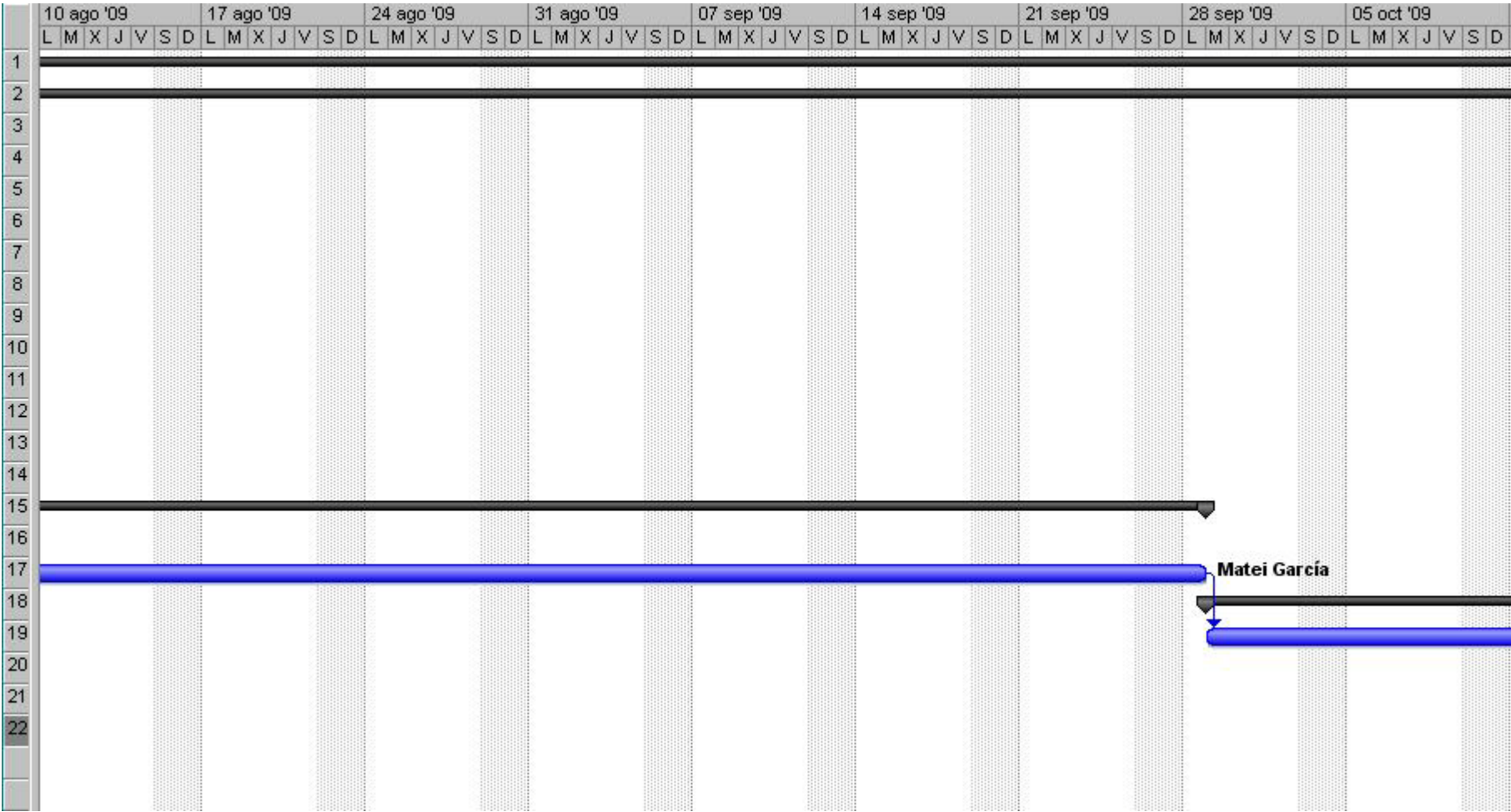
A continuación mostramos el gantt del proyecto. Debido a su extensión, se mostrará en varias páginas

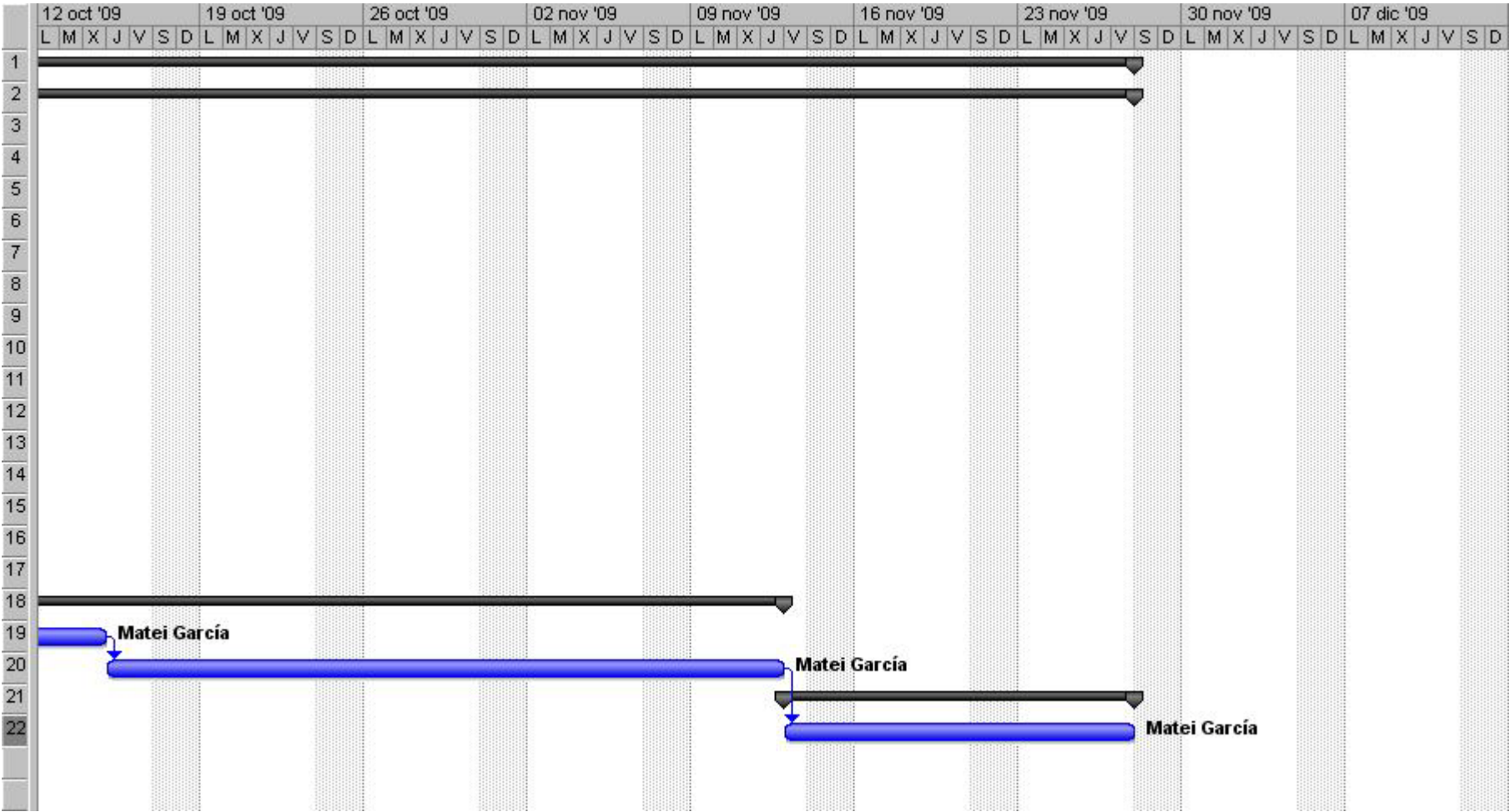
<b>Compañía</b>	Universidad Carlos III de Madrid
<b>Fecha actual</b>	11/12/2009
<b>Título</b>	Proyecto_FC
<b>Comienzo del proyecto</b>	15/4/2009 9:00:00 a.m.
<b>Fin del proyecto</b>	27/11/2009 7:00:00 p.m.













## Anexo 1.2 Gantt del Proyecto Tareas

A continuación mostramos las tareas del gantt del proyecto, junto con su duración, fecha de inicio, fecha de fin, predecesoras y recursos asignados

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1	<input type="checkbox"/> <b>Proyecto Sistema Intermediario</b>	<b>163 días</b>	<b>mié 15/04/09</b>	<b>vie 27/11/09</b>		
2	<input type="checkbox"/> <b>Comienzo del proyecto</b>	<b>163 días</b>	<b>mié 15/04/09</b>	<b>vie 27/11/09</b>		
3	<input type="checkbox"/> <b>Fase de Análisis</b>	<b>56 días</b>	<b>mié 15/04/09</b>	<b>mié 01/07/09</b>		
4	Reunión con el tutor del proyecto	1 día	mié 15/04/09	mié 15/04/09		Matei García; Jose María Sierra
5	Especificación Inicial de Requisitos	10 días	jue 16/04/09	mié 29/04/09	4	Matei García
6	Reunión con el tutor del proyecto	1 día	jue 30/04/09	jue 30/04/09	5	Matei García; Jose María Sierra
7	Validación del diseño preliminar	3 días	mié 13/05/09	vie 15/05/09	12	Jose María Sierra
8	Validación del prototipo	3 días	lun 15/06/09	mié 17/06/09	16	Jose María Sierra
9	Especificación Final de requisitos	7 días	jue 18/06/09	vie 26/06/09	8	Matei García; Jose María Sierra
10	Validación de la Especificación Final de Req.	3 días	lun 29/06/09	mié 01/07/09	9	Jose María Sierra
11	<input type="checkbox"/> <b>Fase de Diseño</b>	<b>66 días</b>	<b>vie 01/05/09</b>	<b>vie 31/07/09</b>		
12	Diseño preliminar del Sistema	8 días	vie 01/05/09	mar 12/05/09	6	Matei García
13	Diseño de prototipo del sistema	6 días	lun 18/05/09	lun 25/05/09	7	Matei García
14	Diseño del proyecto	22 días	jue 02/07/09	vie 31/07/09	10	Matei García
15	<input type="checkbox"/> <b>Fase de Implementación</b>	<b>90 días</b>	<b>mar 26/05/09</b>	<b>lun 28/09/09</b>		
16	Implementación del prototipo del sistema	14 días	mar 26/05/09	vie 12/06/09	13	Matei García
17	Implementación del proyecto	41 días	lun 03/08/09	lun 28/09/09	14	Matei García
18	<input type="checkbox"/> <b>Fase de Pruebas</b>	<b>33 días</b>	<b>mar 29/09/09</b>	<b>jue 12/11/09</b>		
19	Creación de documento de Pruebas	12 días	mar 29/09/09	mié 14/10/09	17	Matei García
20	Realización de Pruebas del sistema	21 días	jue 15/10/09	jue 12/11/09	19	Matei García
21	<input type="checkbox"/> <b>Documentación del sistema</b>	<b>11 días</b>	<b>vie 13/11/09</b>	<b>vie 27/11/09</b>		
22	Realización de la Memoria del Sistema	11 días	vie 13/11/09	vie 27/11/09	20	Matei García



### Anexo 1.3 Presupuesto del Proyecto

Se incluyen a continuación las tablas de presupuesto del proyecto.

Presupuesto Material				
Número	Producto	Cantidad	Coste	Coste Total
1	Ordenador	2	1500	3000
2	Microsoft Windows	2	70	140
<b>TOTAL</b>				3140€

Personal Necesario/Més				
Tipo	Número	Coste	Nº meses	Coste total
Programador	1	1000	3	3000
Analista	1	1600	4	6400
Jefe de proyecto	1	2500	1	2500
<b>TOTAL</b>				11900€

Coste total del proyecto	
<b>Presupuesto Material</b>	3140€
<b>Número</b>	12900€
<b>TOTAL PROYECTO</b>	16040€



## ANEXO 2 – MANUAL DE USUARIO

Debido a la complejidad de la Aplicación vamos a realizar un manual de usuario para ayudar al usuario

Pantalla principal:

Rembrandt Inter-Museos

Consultar Obras

Modificar Pedido

Solicitar Pedido

Ver Pedidos Realizados

Desconectar

Sistema Intermediario para préstamo de Obras de Arte

Por favor auténtíquese

Usuario

Clave\*\*

Login

[Información de Contacto](#)

Esta es la pantalla de presentación del sistema. Tal como hemos descrito antes, la autenticación se produce en la parte superior derecha. El usuario deberá introducir su nombre de usuario y contraseña, y pulsar el botón login.

### Usuario Registrado

A continuación vamos a describir cada una de las funcionalidades del sistema que puede realizar un usuario registrado en el mismo.

Para ello, utilizaremos una serie de ejemplos gráficos para aclarar la forma de acceder y utilizar el sistema.

En el menú de usuario registrado aparecerán cuatro opciones. Vamos a ir indicando una por una la información que nos aportan y como realizar solicitudes al sistema.



Consulta de una obra de arte por parte de un Usuario Registrado:

Rembrandt Inter-Museos

Consultar Obras

Modificar Pedido

Solicitar Pedido

Ver Pedidos Realizados

Desconectar

Introduzca aquí su consulta

Consultar

Bienvenido al Sistema

Está usted conectado/a con el usuario admin

Mail Actual: pepe@marduk1.com

Teléfono Actual: 988777666

Cambiar Datos

Información de Contacto

http://localhost:48232/Proyecto\_FC/ServletControlador?opc=a

Internet | Modo protegido: activado

100%

Para realizar una consulta, deberá introducir un fragmento del título de la obra o del autor de la misma y pulsar “Consultar”.

Rembrandt Inter-Museos

Consultar Obras

Modificar Pedido

Solicitar Pedido

Ver Pedidos Realizados

Desconectar

Resultado de su consulta

Seleccionar	ID Obra	Fecha	Título	Autor	Prestado
<input type="checkbox"/>	AA33333	4/10/1911	Dulcinea	Marcel Duchamp	23/12/2010
<input type="checkbox"/>	AA33334	4/10/1912	Mona Lisa	Marcel Duchamp	NO
<input type="checkbox"/>	AA33334	4/10/1911	Sonata	Marcel Duchamp	NO
<input type="checkbox"/>	AA33330	4/10/1910	Cuadro con basura	Marcel Duchamp	NO

Añadir a Pedido

Volver

Bienvenido al Sistema

Está usted conectado/a con el usuario admin

Mail Actual: pepe@marduk1.com

Teléfono Actual: 988777666

Cambiar Datos

Información de Contacto

Internet | Modo protegido: activado

100%

Nos saldrá una tabla mostrando todos aquellos resultados que coincidan con el patrón indicado.

Si quisiéramos más información sobre una determinada obra de arte, únicamente tendríamos que pinchar en su identificador y obtendríamos la siguiente información:





**Rembrandt Inter-Museos**

Consultar Obras

Modificar Pedido

Solicitar Pedido

Ver Pedidos Realizados

Desconectar

### Información Adicional

ID Obra	Nombre Prestador	Técnica	Medidas	Dirección de Recogida	Dirección de Devolución	Valor del seguro	Observaciones del seguro	Correo
AA33330	Museo Metropolitano de N.Y	Oleo sobre tela	146x114	nada	nada	100000	nada	NO

[Volver](#)

**Bienvenido al Sistema**

Está usted conectado/a con el usuario admin

**Mail Actual:**  
pepe@marduk1.com

**Teléfono Actual:**  
988777666

Cambiar Datos

[Información de Contacto](#)

Internet | Modo protegido: activado
100%

Es decir, mayor información acerca de una determinada obra de arte. Para volver al menú anterior, simplemente pulsar volver.

A continuación, mostraremos como solicitar un pedido al sistema.

**Rembrandt Inter-Museos**

Consultar Obras

Modificar Pedido

Solicitar Pedido

Ver Pedidos Realizados

Desconectar

### Resultado de su consulta

Seleccionar	ID Obra	Fecha	Título	Autor	Prestado
<input type="checkbox"/>	AA33330	4/10/1911	Dulcinea	Marcel Duchamp	23/12/2010
<input checked="" type="checkbox"/>	AA33331	4/10/1912	Mona Lisa	Marcel Duchamp	NO
<input type="checkbox"/>	AA33332	4/10/1911	Sonata	Marcel Duchamp	NO
<input type="checkbox"/>	AA33333	4/10/1910	Cuadro con basura	Marcel Duchamp	NO

Añadir a Pedido

Volver

**Bienvenido al Sistema**

Está usted conectado/a con el usuario admin

**Mail Actual:**  
pepe@marduk1.com

**Teléfono Actual:**  
988777666

Cambiar Datos

[Información de Contacto](#)

Internet | Modo protegido: activado
100%

En la pantalla de consulta, seleccionamos todas las obras que nos interesen y pulsamos sobre “Añadir a Pedidos”.

A continuación nos aparecerá una pantalla como la siguiente:



**Rembrandt Inter-Museos**

Consultar Obras

Modificar Pedido

Solicitar Pedido

Ver Pedidos Realizados

Desconectar

### Obras de arte en su pedido

Seleccionar	Año	Título	Autor	Prestado
<input checked="" type="checkbox"/>	4/10/1912	Mona Lisa	Marcel Duchamp	NO

Solicitar

Eliminar

Consultar mas Obras

**Bienvenido al Sistema**

Está usted conectado/a con el usuario admin

Mail Actual:  
pepe@marduk1.com

Teléfono Actual:  
988777666

Cambiar Datos

[Información de Contacto](#)

Internet | Modo protegido: activado
100%

En esta pantalla nos aparecen todas las obras que están actualmente en nuestro pedido. En este ejemplo vamos únicamente a solicitar una obra, si quisiéramos solicitar más, pulsaríamos sobre la opción “Consultar mas Obras”, que nos llevaría a la pantalla de consultas donde podríamos repetir la operación anterior. A esta pantalla actual, también podremos acceder mediante la opción “Modificar Pedido”.

Otra de las opciones que tenemos es eliminar una de las obras de nuestro pedido. Si quisiéramos hacerlo, solo deberíamos seleccionar las obras que no nos interese solicitar y pulsar sobre el botón “Eliminar”.

Cuando hayamos solicitado todas las obras que deseemos, pulsaremos sobre el botón “Solicitar”.

**Rembrandt Inter-Museos**

Consultar Obras

Modificar Pedido

Solicitar Pedido

Ver Pedidos Realizados

Desconectar

### Por favor indique las fechas en las que desea tener prestada la obra

Seleccionar	Año	Título	Autor	Prestado	Fecha Prestamo	Fecha Devolucion
<input checked="" type="checkbox"/>	4/10/1912	Mona Lisa	Marcel Duchamp	NO		

Solicitar

Volver

**Bienvenido al Sistema**

Está usted conectado/a con el usuario admin

Mail Actual:  
pepe@marduk1.com

Teléfono Actual:  
988777666

Cambiar Datos

[Información de Contacto](#)

Internet | Modo protegido: activado
100%

A continuación, nos aparecerá la pantalla de Solicitud. En dicha pantalla deberemos introducir las fechas entre las cuales estará comprendido el préstamo de la obra.

**NOTA:** Como mínimo la fecha de inicio debe ser diez días naturales a partir del día actual, para que el prestador tenga tiempo de entregar la obra.

Por último pulsaremos el botón Solicitar, y la solicitud se realizará.



Cabe destacar, que en cualquier momento podremos acceder a la pantalla de Solicitud mediante la opción “Solicitar Pedido”

**Rembrandt Inter-Museos**

Consultar Obras

Modificar Pedido

Solicitar Pedido

Ver Pedidos Realizados

Desconectar

**Por favor indique las fechas en las que desea tener prestada la obra**

Seleccionar	Año	Título	Autor	Prestado	Fecha Prestamo	Fecha Devolucion
<input checked="" type="checkbox"/>	4/10/1912	Mona Lisa	Marcel Duchamp	NO		

Solicitar

Volver

< 4/10/2010 >

Lu	Ma	Mi	Ju	Vi	Sa	Do
13	20	21	1	2	3	4
14	5	6	7	8	9	10
15	12	13	14	15	16	17
16	19	20	21	22	23	24
17	26	27	28	29	30	1
18	3	4	5	6	7	8

Limpia Cierre

**Bienvenido al Sistema**

Está usted conectado/a con el usuario admin

Mail Actual:  
pepe@marduk1.com

Teléfono Actual:  
988777666

Cambiar Datos

[Información de Contacto](#)

Listo Internet | Modo protegido: activado 100%

En cualquier momento, tenemos la posibilidad de consultar todos los pedidos realizados y el estado de los mismos. Para ello, simplemente tendremos que pulsar la opción “ver pedidos realizados” en el menú izquierdo.

**Rembrandt Inter-Museos**

Consultar Obras

Modificar Pedido

Solicitar Pedido

Ver Pedidos Realizados

Desconectar

**Resultado de su consulta**

Obra	Pedido	Título	Usuario	Inicio de Prestamo	Fin de Prestamo	Validado
AA33333	28	Dulcinea	00000000H	17/1/2010	17/1/2014	SI
AA33333	29	Dulcinea	00000000H	24/1/2010	18/9/2012	SI
AA33333	30	Dulcinea	00000000H	24/1/2010	6/9/2010	SI
AA33333	31	Dulcinea	00000000H	24/1/2010	6/1/2013	SI
AA33333	32	Dulcinea	00000000H	24/1/2010	10/1/2015	FF
AA33332	32	Mona Lisa	00000000H	24/1/2010	12/1/2014	SI
AA33334	32	Sonata	00000000H	24/1/2010	11/1/2014	FF
AA33330	32	Cuadro con basura	00000000H	24/1/2010	20/1/2014	FF

**Bienvenido al Sistema**

Está usted conectado/a con el usuario admin

Mail Actual:  
pepe@marduk1.com

Teléfono Actual:  
988777666

Cambiar Datos

[Información de Contacto](#)

Listo Internet | Modo protegido: activado 100%



## Validador

A continuación vamos a describir cada una de las funcionalidades del sistema que puede realizar validador registrado en el mismo.

Para ello, tal como hicimos con el usuario registrado, utilizaremos una serie de ejemplos gráficos para aclarar la forma de acceder y utilizar el sistema.

Para validar un pedido, seleccionaremos la opción validar pedidos de la pestaña izquierda.

**Rembrandt Inter-Museos**

**Validar Pedidos**

**Mostrar Pedidos**

**Desconectar**

**Resultado de su consulta**

Seleccionar	Obra	Pedido	Usuario	Fecha Inicio	Fecha Fin
<input type="checkbox"/>	AA33332	40	marduki1	14/4/2010	11/2/2015

**Validar** **Negar**

**Está usted conectado/a con el usuario valid**

**Mail Actual:**  
valid@marduki.es

**Teléfono Actual:**  
999999999

**Cambiar Datos**

[Información de Contacto](#)

Internet | Modo protegido: activado 100%

Si lo desea el validador puede solicitar más información al sistema acerca del usuario que ha realizado el pedido. Para ello simplemente deberá pulsar en el ID usuario correspondiente al pedido sobre cuyo usuario desea informarse.

**Rembrandt Inter-Museos**

**Validar Pedidos**

**Mostrar Pedidos**

**Desconectar**

**Información Adicional**

NIF Usuario	Nombre	Mail	Dirección	Teléfono
00000000H	admin	pepe@marduki.com	admin	988777666

**Volver**

**Está usted conectado/a con el usuario valid**

**Mail Actual:**  
valid@marduki.es

**Teléfono Actual:**  
999999999

**Cambiar Datos**

[Información de Contacto](#)

Internet | Modo protegido: activado 100%



A continuación seleccionaremos todas aquellas solicitudes que deseemos validar y pulsaremos el botón Validar.

Proyecto Sistema Intermediario		
<div>Validar Pedidos</div> <div>Mostrar Pedidos</div> <div>Desconectar</div>	<div>PEDIDO VALIDADO </div> <div>Volver</div>	<div>Bienvenido al Sistema</div> <div>Está usted conectado/a con el usuario valid</div> <div>Mail Actual: validadores@val.es</div> <div>Teléfono Actual: 999999888</div> <div>Cambiar Datos</div>
<div><a href="#">Información de Contacto</a></div>		

Internet | Modo protegido: activado 100%

De igual manera, podemos negar un pedido, seleccionando los pedidos que deseemos negar, y pulsando sobre la opción “Negar”.

Proyecto Sistema Intermediario		
<div>Validar Pedidos</div> <div>Mostrar Pedidos</div> <div>Desconectar</div>	<div>PEDIDO NEGADO </div> <div>Volver</div>	<div>Bienvenido al Sistema</div> <div>Está usted conectado/a con el usuario valid</div> <div>Mail Actual: validadores@val.es</div> <div>Teléfono Actual: 999999888</div> <div>Cambiar Datos</div>
<div><a href="#">Información de Contacto</a></div>		

Listo Internet | Modo protegido: activado 100%



Además podremos consultar todos los pedidos validados, negados o pendientes de validar. Para ello utilizaremos la opción Mostrar Pedidos.

**Rembrandt Inter-Museos**

Validar Pedidos

Mostrar Pedidos

Desconectar

**Resultado de su consulta**

Obra	Pedido	Usuario	Fecha Inicio	Fecha Fin	Validado
AA33333	28	oooooooooH	17/1/2010	17/1/2014	SI
AA33333	29	oooooooooH	24/1/2010	18/9/2012	SI
AA33333	30	oooooooooH	24/1/2010	6/9/2010	SI
AA33333	31	oooooooooH	24/1/2010	6/1/2013	SI
AA33333	32	oooooooooH	24/1/2010	10/1/2015	FF
AA33332	32	oooooooooH	24/1/2010	12/1/2014	SI
AA33334	32	oooooooooH	24/1/2010	11/1/2014	FF
AA33330	32	oooooooooH	24/1/2010	20/1/2014	FF

Está usted conectado/a con el usuario valid

Mail Actual:  
valid@marduk1.es

Teléfono Actual:  
999999999

Cambiar Datos

[Información de Contacto](#)

http://localhost:48232/Proyecto\_FC/ServletControlador?opc=k
 Internet | Modo protegido: activado 100%

Indicar también que, tanto un usuario registrado como un validador, tienen la opción de cambiar su teléfono y mail en el sistema.

Para ello deberán pulsar sobre el botón Cambiar Datos de su menú lateral derecho.

**Rembrandt Inter-Museos**

Consultar Obras

Modificar Pedido

Solicitar Pedido

Ver Pedidos Realizados

Desconectar



**Sistema Intermediario para préstamo de Obras de Arte**

**Bienvenido al Sistema**

Está usted conectado/a con el usuario admin

Mail Actual:  
pepe@marduk1.com

Teléfono Actual:  
999666333

Cambiar Datos

[Información de Contacto](#)

Listo
 Internet | Modo protegido: activado 100%

A continuación aparecerá una pantalla donde el usuario introducirá los nuevos datos a modificar. Para hacer efectivos dichos cambios deberá pulsar el botón Enviar.

### Rembrandt Inter-Museos

Consultar Obras

Modificar Pedido

Solicitar Pedido

Ver Pedidos Realizados

Desconectar

Introduzca los nuevos datos por favor

pepe@marduk1.com

Mail

988777666

Tel.

Enviar

Cancelar

[Información de Contacto](#)

Y el cambio se realizará en el sistema

### Rembrandt Inter-Museos

Consultar Obras

Modificar Pedido

Solicitar Pedido

Ver Pedidos Realizados

Desconectar

Se han cambiado correctamente sus datos

Volver

[Información de Contacto](#)



Finalmente indicar que el cambio se reflejará en los datos actuales en la parte derecha de la pantalla



Por último mencionar que existe una opción Desconectar en la parte inferior izquierda para que los usuarios y validadores puedan desconectarse del sistema